



Jira Service Management 5.4

Contents

Jira Service Management Data Center and Server 5.4 documentation	6
Installing Jira Service Management	7
Jira applications overview	10
Permissions overview	12
Getting started with Jira Service Management	27
Getting started for service project admins	30
Setting up your service project	32
Creating service project request types	34
Making queues for your service project teams	38
Adding service project agents	41
Customize your service project channels	44
Bring your service project to the next level	48
Introduce customers to your service project	50
Explore a sample project	52
Getting started for service project agents	54
Administering service projects	57
Managing access to your service project	58
Configuring the customer portal	63
Receiving requests by email	68
Managing the email channel	71
Troubleshooting issues with the email channel	72
Setting up service project users	75
Managing project role memberships	81
Setting up queues for your team	82
Automating your service project	84
Send alerts with Jira Service Management webhooks	89
Managing service project notifications	91
Translate service project notifications	99
Setting up request types	103
Troubleshooting issues with request types	108
Setting up SLAs	111
Create and edit SLA calendars	118
How teams see SLAs	123
Reporting on SLAs	130
Importing SLAs	132
Using JQL queries specific to SLAs	135
Example: creating a basic SLA	137
Example: creating an SLA that doesn't track continuous time	138
Example: creating an SLA with multiple cycles	139
Example: creating an SLA based on due date	140
Setting up approvals	141
Setting up service project reports	146
Default service project configuration	153
Using Jira applications with Confluence	156
Working on service projects	159
Working with issues	160
Attaching files and screenshots to issues	162
Creating issues and sub-tasks	165
Creating issues using the CSV importer	168
Editing and collaborating on issues	175
Linking issues	179
Editing multiple issues at the same time	184
Moving an issue	190
Visual editing	191
Scheduling an issue	193
Logging work on issues	195
Approving a service project request	199
Customizing the issues in a project	201

Searching for issues	204
Basic searching	206
Quick searching	208
Advanced searching	212
Advanced searching - fields reference	222
Advanced searching - keywords reference	257
Advanced searching - operators reference	260
Advanced searching - functions reference	271
Search syntax for text fields	296
Saving your search as a filter	302
Working with search results	306
Constructing cron expressions for a filter subscription	315
Managing your user profile	317
Allowing OAuth access	319
Requesting apps	320
Using keyboard shortcuts	321
Adding announcements	323
Adding request participants	326
Using service project queues	330
Raising requests on behalf of customers	333
Adding customers	334
Organizing work with versions	335
Workflows	338
Using Jira on a mobile device	340
Configuring dashboards	341
Adding and customizing gadgets	344
Gadgets for Jira applications	345
Using the help center	352
View requests in the help center	356
Configuring request view	358
Organizing work with components	361
Set up a knowledge base for self-service	363
Set up a knowledge base with Confluence Server	364
Knowledge base settings and permissions	367
Set up a knowledge base with Confluence Cloud	369
Knowledge base settings and permissions in Confluence Cloud	372
Write and search for knowledge base articles	375
Setting up incident management	383
Getting Opsgenie API information	385
Working with major incidents	386
Managing your assets with Assets	389
Getting started with Assets	390
What is Assets?	392
Tutorial: Starting from scratch	398
1. Build your first object schema	399
2. Create object types	401
3. Add attributes	403
4. Create objects	405
5. View object graphs	407
6. Link your object schema to a Jira project	409
7. Add object to your requests	411
Tutorial: How Jira issues affect Assets objects (ITSM)	413
1. Create an ITSM project	414
2. Create an incident	416
3. Work on the problem	420
4. Work on the change	423
Administering Assets	427
Installing Assets	428
Configuring roles and permissions	431
Configuring global Jira settings	434
Configuring global Assets settings	440
Adding Assets custom fields to screens in Jira	442
Default Assets custom field	443
Referenced Assets custom field	446

Read-only Assets custom field	449
Enabling Assets custom fields on customer portals	451
Adding Assets functions to workflows in Jira	453
Conditions and validators	454
Post functions	455
Groovy scripts	460
Adding approvers from Assets to requests in Jira	462
Importing and exporting object schemas	467
Assets REST API documentation	470
Advanced usage guides	471
Upgrading Insight with Jira Service Management	472
Working with object schemas	474
Creating object schemas	475
Object schema templates	477
Configuring object schemas	479
Configuring Assets automation rules	481
Configuring print labels	489
Working with object types	491
Creating object types	493
Adding attributes to object types	494
Configuring inheritance of attributes	499
Customizing the object view layout	500
Adding widgets	502
Working with objects	510
Creating objects	512
Adding objects to Jira issues	514
Viewing objects in the object view	517
Viewing references in a graph	522
Printing labels and QR codes	525
Importing your data	526
Importing with Asset Discovery	527
Importing with Integrations	528
Importing with built-in importers	529
Understand importing concepts	530
1. Create your import configuration	533
2. Create object type and attribute mapping	552
3. Inspect your import configuration	556
4. Enable and sync your import	557
Searching for objects	558
Advanced searching: AQL - Assets Query Language	560
Advanced: Placeholders	573
Assets AQL functions	578
Working with reports	580
Providing help in multiple languages	585
Collecting customer satisfaction (CSAT) feedback	589
Jira Service Management best practices	592
Best practices for designing the customer portal	593
Best practices for IT teams using Jira Service Management	597
Fulfilling service requests with your IT service desk	599
Managing changes with your IT service project	604
What to expect: Sample lifecycle of a change	607
1. Update the change management workflow	652
2. Create resolutions, post-functions, and priorities	654
3. Create and update custom fields	656
4. Link your project to Insight asset management	659
5. Add approvals to your workflow	663
6. Create and update automation rules	666
7. (Optional) Set up a calendar to coordinate your changes	670
Managing incidents with your IT service desk	671
Managing problems with your IT service desk	675
Calculating priority automatically	679
Manage changes with your IT service management project 1	682
Best practices for software teams using Jira Service Management	685
Collaborate with other Jira teams on Jira Service Management issues	686

Collect effective bug reports from customers	688
Customize Jira Service Management's bug report workflow	690
Escalate Jira Service Management issues to other Jira teams	693
Get set up for customer service	694
Getting help with Jira Service Management	696
Automate your project	698
Jira Data Center and Server mobile app	702
Accessibility	706

Jira Service Management Data Center and Server 5.4 documentation

Put the power of Jira in the hands of your service management team.

Get started

New to Jira Service Management? Check out our guides for new administrators and users.

[View guide](#)

Whats new

Time to upgrade? Get the lowdown on the latest and greatest in Jira Service Management 5.4.

[View latest changes](#)

Installing Jira Service Management

Jira Service Management is built on the Jira platform and has everything your IT teams need for service request, incident, problem, and change management. It's part of the Jira family of applications (Jira Software, Jira Service Management, Jira Core) and can be used separately or in any combination, on the same instance.

If you're ready to install or upgrade Jira Service Management, then this guide has you covered. If you have Jira Service Management installed and would like to learn how to configure and use it, head to the [Jira Service Management documentation](#).



1. System requirements

Before you install Jira Service Management, you'll need to check if it supports your operating system, as well as a few other things.

Read

- [Supported platforms](#)
- [Jira applications installation requirements](#)

If everything looks good, it's time to decide which version and license is right for your organization. If you've already made a decision, jump to [Install Jira Service Management](#).



2. Versions and licensing

If you plan on managing your own Jira Service Management instance (not hosted by Atlassian), you have the option of either a Server or Data Center license. Your license determines which features and infrastructure choices are available.

For organizations who need more time to prepare before upgrading to a new version, but still want critical bug fixes, an Enterprise release is a good choice.

Read

- [Jira Service Management licensing and purchasing](#)
- [Jira Server and Data Center feature comparison](#)
- [Long Term Support releases](#)

If you're clear on what you're getting and why, it's time to install Jira Service Management.



3. Install Jira Service Management

Here we'll talk you through how to install Jira Service Management to trial and put into production, as well as how to add additional Jira applications to your existing installation.

Install Jira Service Management to trial

1. Download the installer for your operating system at <https://www.atlassian.com/software/jira/service-desk/download>.
2. Follow the steps at [Evaluation installation](#). Jira Service Management comes with a handy embedded H2 database that you can use for evaluation purposes.
3. Once installed, read [Getting started for service project admins](#) to learn what you can do with Jira Service Management.

If you'd like to move into production, consider starting afresh by following the [Install Jira Service Management for production](#) guides below.

Install Jira Service Management for production

1. Set up your database by following the guide that applies to you at [Connecting Jira applications to a database](#).
2. Install with your operating system by following the guide below that applies to you:
 - [Installing Jira applications on Windows](#)
 - [Installing Jira applications on Linux](#)
3. Once installed, follow the tutorial on [Getting started for service project admins](#) to set up Jira Service Management for your support teams.

Install additional Atlassian products

If you have Jira Core or Jira Software installed, you can install Jira Service Management as an additional application, and vice versa.

Head to **Administration** () > **Applications > Versions & licenses** and follow the steps below:

1. Check which version of Jira Service Management is compatible with your existing installation at [Jira Service Management version history](#).
2. Ensure you have Jira Administrator Global Permission.
3. Follow the steps at [Installing additional applications and version updates](#).
4. Read the [documentation](#) for the product you've installed, to learn how to get started.



There are over 600 apps available from the [Atlassian Marketplace](#) that will help you supercharge your Jira Service Management. Read [Managing apps](#) to learn how to set these up.



4. Upgrade to a newer version

There are several ways to upgrade Jira Service Management. The method you choose will depend on the version of Jira Service Management you use, and the type of environment you use it in.

Steps to take

1. Decide which version to upgrade to by reading the [release and upgrade notes](#).
2. Determine your upgrade path:

Version	Upgrade path
Earlier than 2.5.x	Upgrade to 3.0, then upgrade to the latest version.
3.0 to 3.16	Upgrade directly to the latest version.
4.0 or later	Upgrade directly to the latest version.

3. Choose your method from the below options and follow the steps on that page:
 - [Using the installer](#) (the easiest way to upgrade your Jira instance)
 - [Upgrading manually](#) (if you're moving to a different operating system or database software)
 - [Upgrading Jira Data Center \(manual\)](#) with downtime
 - [Upgrading Jira Data Center](#) with zero downtime



5. Get the most out of Jira Service Management

Now you have Jira Service Management installed, there's plenty of resources to help you set it up, learn the features, and get your dream team using it.

Getting started

- [Getting started for service project admins](#)
- [Getting started for service project agents](#)

Top tasks

- [Configuring the Customer Portal](#)
- [Automating your service project](#)
- [Setting up SLAs](#)

Best practice guides

- [Set up a knowledge base for self-service](#)
- [Best practices for IT teams using Jira Service Management](#)
- [Best practices for software teams using Jira Service Management](#)

Jira applications overview




Jira Service Management licensing overview

The Jira family of applications (Jira Software, Jira Service Management, Jira Core) are built on the Jira platform and can be used in any combination on the same instance. Depending upon of your setup, users can be licensed to one, all, or any combination of these applications. Read on to understand how Jira Service Management licensing and roles affect what agents, customers, and other Jira application users can do.

If you're a Jira administrator, check out more information on [Licensing and application access](#).

Application features and project types

Each application delivers a tailored experience for its users, and has an associated project type which in turn offers application specific features. Below is a list of the project types, and their associated application specific features.

Application	Project type	Application specific feature set
Jira Core	 Business projects	<ul style="list-style-type: none">• Available to all licensed users
Jira Software	 Software projects	<ul style="list-style-type: none">• Integration with development tools• Agile boards• Release hub for software versions
Jira Service Management	 Service projects	<ul style="list-style-type: none">• Service Level Agreements (SLAs)• A customizable web portal for customers• Permission schemes allowing customer access

All users that can log in to a Jira instance will be able to see all the projects in that instance (pending permissions), but they will only be able to see the application-specific features when they have application access. For example, a Software project is able to display information from linked development tools, such as Bitbucket and FishEye, as well as agile boards, but this information is only viewable by a Jira Software user. A Jira Core user would be able to see the Software project, but would not be able to see the Software-specific features, like agile boards or the information from linked development tools. Likewise, a Jira Software user would not be able to see any Jira Service Management application-specific features on a service project, only a basic view of the project and its issues.



- Only a Jira administrator can create a project for an installed application. They do not need application access to create the project, but they do need application access if they'd like to view or use the project.
- Anonymous users will have access equivalent to Jira Core users. In other words, they can view issues and work in any type of project, but they won't see application-specific features, e.g. agile boards, which are Jira Software-specific features. To know how to allow anonymous users access to projects, see [Allowing anonymous access to your instance](#).

A list of the applications, their user roles, and their project's application-specific features can be found below:

			Jira Core	Jira Software	Jira Service Management
--	--	--	-----------	---------------	-------------------------

			Jira-Core-user	Jira-Software-user	Jira-ServiceDesk-agent	Customers
Service Projects 	Project level	Create	✗	✗	✓	✗
		View	✓	✓	✓	✓
	Issue level	Create	✗	✗	✓	✓
		View	✓	✓	✓	✓
		Comment	✓	✓	✓	✓
		Transition	✗	✗	✓	✗
	SLA level	Create	✗	✗	✓	✗
		View	✗	✗	✓	✗
	Queue level	Create	✗	✗	✓	✗
		View	✗	✗	✓	✗
	Jira Service Management gadgets	View	✗	✗	✓	✗

Permissions overview

This page describes the different types of permissions and access rights that can be set up in Jira applications.

What are permissions?

Permissions are settings within Jira applications that control what users within those applications can see and do. All Jira applications allow a variety of permissions: from whether users can create new projects to whether a user can see a specific type of comment on an issue. These permissions can differ between applications.

Permissions are different from application access, which is controlled by groups that have **Use** access for an application. For more information about setting application access, see [Managing user access to JIRA applications](#).

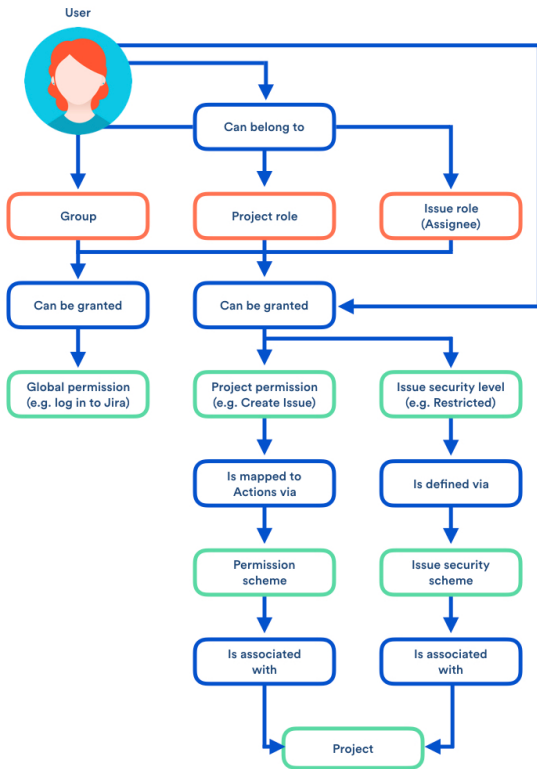
Types of permissions

There are three types of permissions in Jira applications, and they range from the high-level to granular:

- **Global permissions** - These apply to applications as a whole, not individual projects (for example, whether users can see the other users in the application).
- **Project permissions** - Organized into permission schemes, these apply to projects (e.g. who can see the project's issues, create, edit and assign them). While project admins can assign users to a project, they can't customize the permission schemes for a project. There are lots of project-level permissions you can set to control what users can do within a project.
- **Issue security permissions** - Organized into security schemes, these allow the visibility of individual issues to be adjusted (within the bounds of the project's permissions). For example, issue security permissions can let you set up types of issues that can only be seen by project admins or users in specific groups.

How do permissions get assigned?

Permissions can be assigned to groups or to project roles/and or issue roles. This diagram illustrates how permissions are assigned to users:



Who can set permissions?

Permission	Can be set by	For more info, see...
Global permission	A user with the Jira System administrator permission A user in a group with System Admin access	Managing global permissions
Project permission	A user with the Jira System administrator permission A user in a group with Admin access	Managing project permissions
Issue security permission	A user with the Jira System administrator permission A user in a group with Admin access A project admin	Configuring issue-level security

Jira Service Management global and project permissions

Jira Service Management provides a standard permission scheme (Jira Service Management Permission scheme for *project*) that automatically gives your service desk users the correct permissions for the project role they are in. For example, adding agents to your service desk will add users to the Service Desk Team role. This role gives them access to Jira Service Management projects to which they're assigned and also allows them to work on issues.

Global permissions

At installation time, Jira Service Management creates a global permission named **Jira Service Management agent access**. If agent based pricing is enabled for the instance, users who require access to agent views or functionality need to have this permission. The number of users who are granted this permission determines how many agent licenses are used on the system.

Project permissions

This table shows the permission configuration for a standard service desk project permission scheme:

Project Permissions	Users / Groups / Project roles	Explanation
Administer Projects	Project Role (Administrators)	Permission to administer a project. This includes the ability to edit project role membership, project components, project versions and certain project details (Project Name, URL, Project Lead, Project Description).
Browse Projects	<ul style="list-style-type: none"> Service Desk Customer - Portal Access Project Role (Service Desk Team) Project Role (Administrators) 	Permission to browse projects, use the Issue Navigator and view individual issues (except issues that have been restricted via issue security). Many other permissions are dependent on this permission , e.g. the 'Work On Issues' permission is only effective for users who also have the 'Browse Projects' permission.
View Development Tools	<ul style="list-style-type: none"> Project Role (Administrators) 	

View (Read-Only) Workflow	<ul style="list-style-type: none"> • Project Role (Service Desk Team) • Project Role (Administrators) 	Permission to view the project's 'read-only' workflow when viewing an issue. This permission provides the 'View Workflow' link against the Status field of the 'View Issue' page .
Issue Permissions	Users / Groups / Project roles	Explanation
Create Issues	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	Permission to create issues in the project. (Note that the Create Attachments permission is required in order to create attachments.) Includes the ability to create sub-tasks (if sub-tasks are enabled).

<p>Edit Issues</p>	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to edit issues (excluding the 'Due Date' field see the Schedule Issues permission). Includes the ability to convert issues to sub-tasks and vice versa (if sub-tasks are enabled). Note that the Delete Issue permission is required in order to delete issues. The Edit Issue permission is usually given to any groups or project roles who have the Create Issue permission (perhaps the only exception to this is if you give everyone the ability to create issues it may not be appropriate to give everyone the ability to edit too). Note that all edits are recorded in the issue change history for audit purposes.</p>
<p>Transition Issues</p>	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to transition (change) the status of an issue.</p>

<p>Schedule Issues</p>	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to schedule an issue that is, to edit the 'Due Date' of an issue. In older versions of Jira this also controlled the permission to view the 'Due Date' of an issue.</p>
<p>Move Issues</p>	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to move issues from one project to another, or from one workflow to another workflow within the same project. Note that a user can only move issues to a project for which they have Create Issue permission.</p>

<p>Assign Issues</p>	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to assign issues to users. Also allows autocompletion of users in the Assign Issue drop-down. (See also Assignable User permission below)</p>
<p>Assignable User</p>	<ul style="list-style-type: none"> • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to be assigned issues. (Note that this does not include the ability to assign issues; see Assign Issue permission).</p>
<p>Resolve Issues</p>	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to resolve and reopen issues. This also includes the ability to set the 'Fix For version' field for issues. Also see the Close Issues permission.</p>

<p>Close Issues</p>	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to close issues. (This permission is useful where, for example, developers resolve issues and testers close them). Also see the Resolve Issues permission.</p>
<p>Modify Reporter</p>	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to modify the 'Reporter' of an issue. This allows a user to create issues 'on behalf of' someone else. This permission should generally only be granted to administrators.</p>

<p>Delete Issues</p>	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to delete issues. Think carefully about which groups or project roles you assign this permission to; usually it will only be given to administrators. Note that deleting an issue will delete all of its comments and attachments, even if the user does not have the Delete Comments or Delete Attachments permissions. However, the Delete Issues permission does not include the ability to delete individual comments or attachments.</p>
<p>Link Issues</p>	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to link issues together. (Only relevant if Issue Linking is enabled).</p>

<p>Set Issue Security</p>	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to set the security level on an issue to control who can access the issue. Only relevant if issue security has been enabled.</p>
<p>Voters & Watchers Permissions</p>	<p>Users / Groups / Project Roles</p>	<p>Explanation</p>
<p>View Voters and Watchers</p>	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to view the voter list and watcher list of an issue. Also, see the Manage Watcher List permission.</p>

<p>Manage Watcher List</p>	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to manage (i.e. view/add/remove users to/from) the watcher list of an issue.</p>
<p>Comments Permissions</p>		<p>Explanation</p>
<p>Add Comments</p>	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to add comments to issues. Note that this does not include the ability to edit or delete comments.</p>

<p>Edit All Comments</p>	<ul style="list-style-type: none"> • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to edit any comments, regardless of who added them.</p>
<p>Edit Own Comments</p>	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to edit comments that were added by the user.</p>
<p>Delete All Comments</p>	<ul style="list-style-type: none"> • Project Role (Service Desk Team) • Project Role (Administrators) 	<p>Permission to delete any comments, regardless of who added them.</p>

Delete Own Comments	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	Permission to delete comments that were added by the user.
Attachments Permissions	Users / Groups / Project Roles	Explanation
Create Attachments	<ul style="list-style-type: none"> • Service Desk Customer - Portal Access • Project Role (Service Desk Team) • Project Role (Administrators) 	Permission to attach files to an issue. (Only relevant if attachments are enabled). Note that this does not include the ability to delete attachments.

Delete All Attachments	<ul style="list-style-type: none"> Project Role (Service Desk Team) Project Role (Administrators) 	Permission to delete any attachments, regardless of who added them.
Delete Own Attachments	<ul style="list-style-type: none"> Service Desk Customer - Portal Access Project Role (Service Desk Team) Project Role (Administrators) 	Permission to delete attachments that were added by the user.
Time Tracking Permissions	Users / Groups / Project Roles	Explanation
Work On Issues	<ul style="list-style-type: none"> Project Role (Service Desk Team) Project Role (Administrators) 	Permission to log work against an issue, i.e. create a worklog entry. (Only relevant if Time Tracking is enabled).

Edit Own Worklogs	<ul style="list-style-type: none"> Project Role (Service Desk Team) Project Role (Administrators) 	Permission to edit worklog entries that were added by the user. (Only relevant if Time Tracking is enabled). Also, see the Work On Issues permission.
Edit All Worklogs	<ul style="list-style-type: none"> Project Role (Administrators) 	Permission to edit any worklog entries, regardless of who added them. (Only relevant if Time Tracking is enabled). Also, see the Work On Issues permission.
Delete Own Worklogs	<ul style="list-style-type: none"> Project Role (Service Desk Team) Project Role (Administrators) 	Permission to delete worklog entries that were added by the user. (Only relevant if Time Tracking is enabled). Also, see the Work On Issues permission.
Delete All Worklogs	<ul style="list-style-type: none"> Project Role (Administrators) 	Permission to delete any worklog entries, regardless of who added them. (Only relevant if Time Tracking is enabled). Also, see the Work On Issues permission.

Using custom permission schemes

If you are a service desk administrator and you want to customize the standard permission scheme, make sure that the roles have the mandatory permissions. See [Customizing Jira Service Management permissions](#).

Resolving permission scheme errors

If you encounter any error messages related to your service desk's permission scheme, check out [Resolving Jira Service Management permission errors](#).

Getting started with Jira Service Management

Learn about Jira Service Management's different user types and roles, get a brief introduction to how your customers raise requests, and learn what those requests look like for agents. When you're ready, pick a tutorial and learn how to make your service desk work better for your team and your customers.

User types and roles

Licensed users manage customer requests in your instance of Jira Service Management. These are most likely your project administrators, service desk team members, employees and even contractors. They track and resolve your customers' requests, transition issues through workflows, and contribute to their team's service level agreements (SLAs).

Unlicensed users raise requests and interact with service desk agents *for free*. These are your customers. They submit requests via your service project's customer portal or email address, comment on requests, and read knowledge base articles. Customers do not require a license, so you can have an unlimited number of customers who can submit an unlimited number of requests, free of charge.

Most of the information in these documents focuses on two licensed roles: administrators and agents. The administrator sets up and configures service projects. The agent works in these projects.

Admins

Project administrators for your service project can:

- Access all features in Jira Service Management
- Manage users and roles in service projects
- Set up customer portals, request types, queues, reports and SLAs
- Perform all tasks that agents can

Agents

Service project team members who work on customer requests can:

- View the customer portal, queues, reports and SLA metrics for a project
- Add, edit, and delete customer-facing and internal comments on issues
- Add customers to a project
- Read knowledge base content
- Manage organizations (if allowed at the application level)
- Create knowledge base content (with a Confluence license)

Customers





The people you serve through your service project can:

- Raise requests via the customer portal or email channel
- Track their requests in the customer portal
- Comment on their requests
- Read knowledge base articles
- Approve other customers' requests
- Share requests with other customers (if allowed by [Customer permissions](#))

All this is free for your customers.

How a service project works (in 4 simple steps)

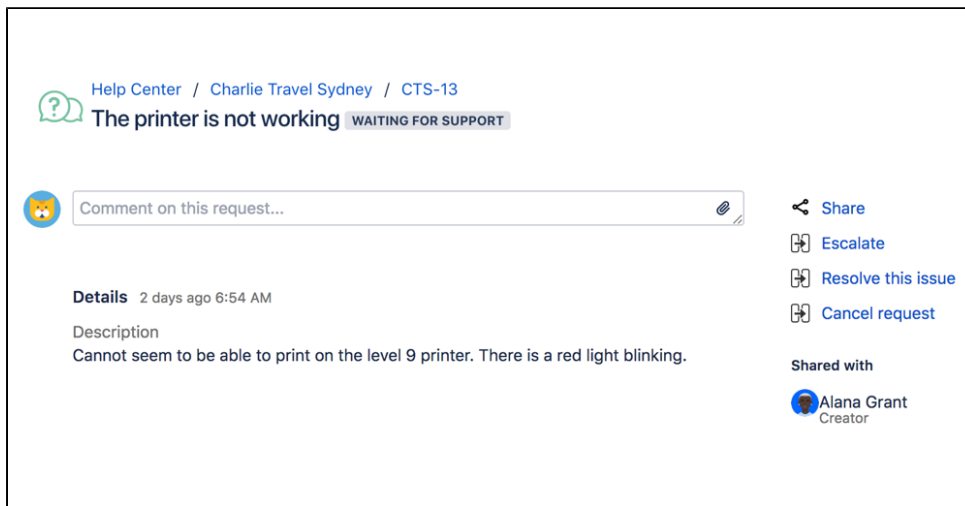
Here's how your customers and service project agents work together to resolve a request:

	<p>1. Your customer submits a request to your service agents through a portal or via email.</p>
	<p>2. A service project agent sees the request in their Jira Service Management queue and looks into the issue.</p>
	<p>3. Your customer and other participants use the portal or email to discuss the request with your service project agent, who works in Jira Service Management.</p>
	<p>4. Your agent completes the request and your customer is satisfied!</p>

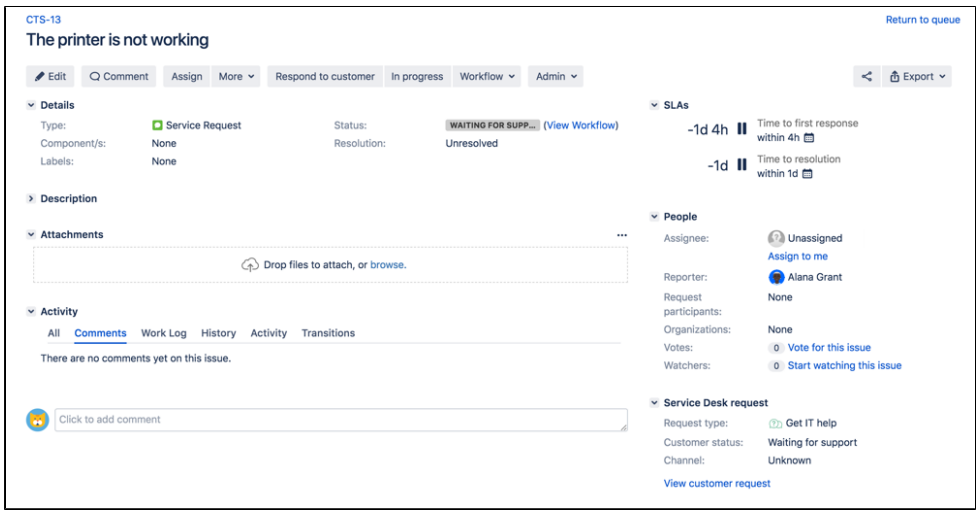
"Requests" vs. "issues"

Your customers submit a *request* through the customer portal or by email. These requests become *issues* that your agents work on in their agent view.

How customers see their *requests* in the customer portal:



How agents see those corresponding *issues*:



Ready to get started?

Choose your Jira Service Management tutorial:

[I am an admin](#)

[I am an agent](#)

Getting started for service project admins

Welcome to Jira Service Management for admins! In this tutorial, we'll introduce you to your workspace and walk you through the process of setting up a service project for your team of agents and a corresponding customer-facing site (which we call the customer portal).

We'll be focusing on basic Jira Service Management features and tasks to help you get up and running quickly. By the end of this tutorial, you will have:

- Set up 1 service project
- Added 3 agents
- Prepared your customer portal to receive customer requests

Audience:

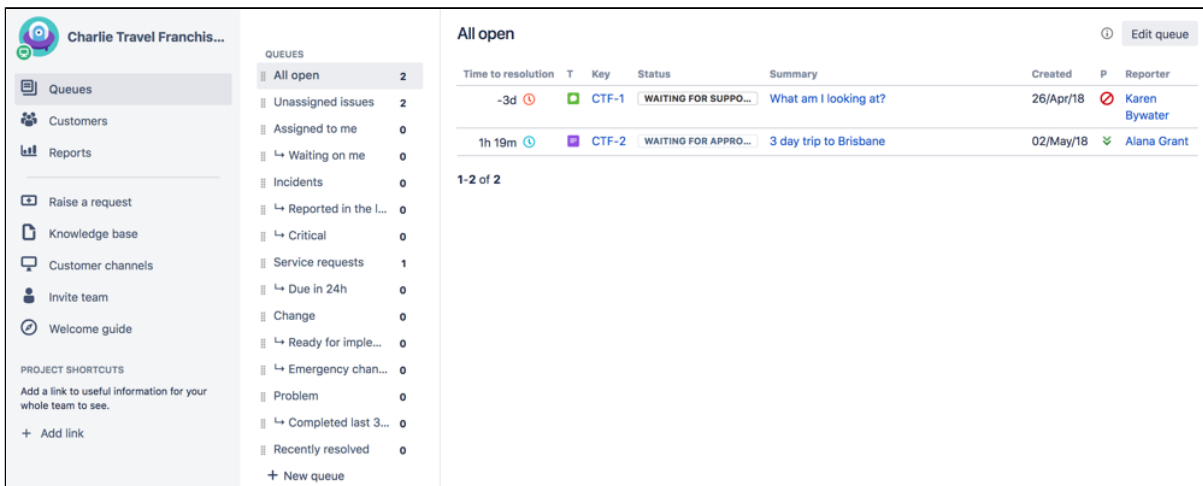
- Service desk administrators
- Team managers

Time: 30 minutes

Let's take a quick look at Jira Service Management...

Queues

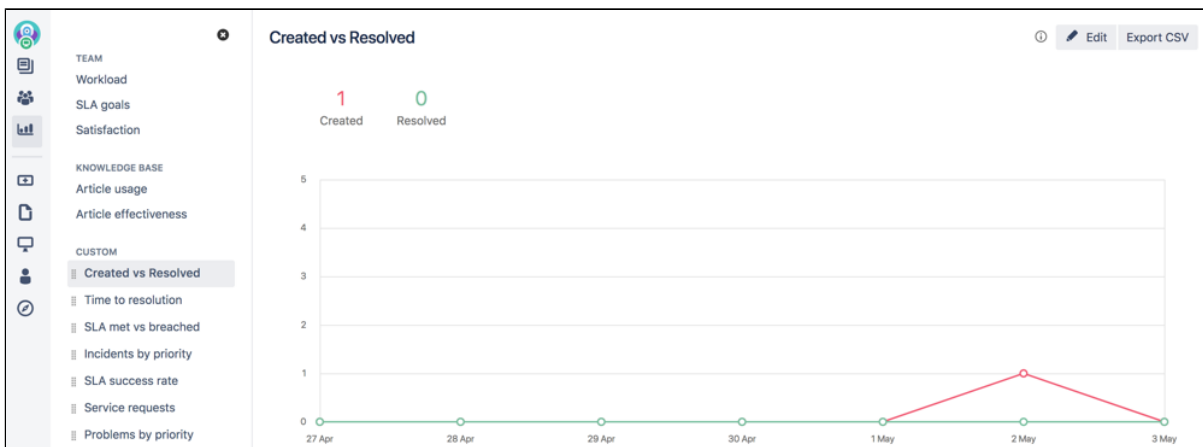
As an admin, you will set up and configure queues for your agents. Your agents will then view and work on issues from the same tab:



Time to resolution	T	Key	Status	Summary	Created	P	Reporter
-3d	🔴	CTF-1	WAITING FOR SUPPO...	What am I looking at?	26/Apr/18	🚫	Karen Bywater
1h 19m	🕒	CTF-2	WAITING FOR APPRO...	3 day trip to Brisbane	02/May/18	✅	Alana Grant

Reports

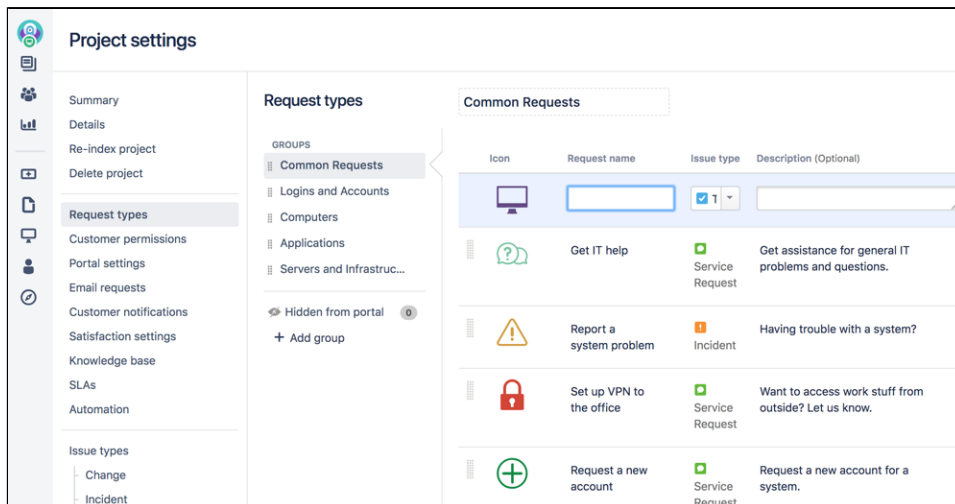
Use the Reports tab to view your team's workload. You can also set up custom reports to track your team's progress in more detail:



Project settings



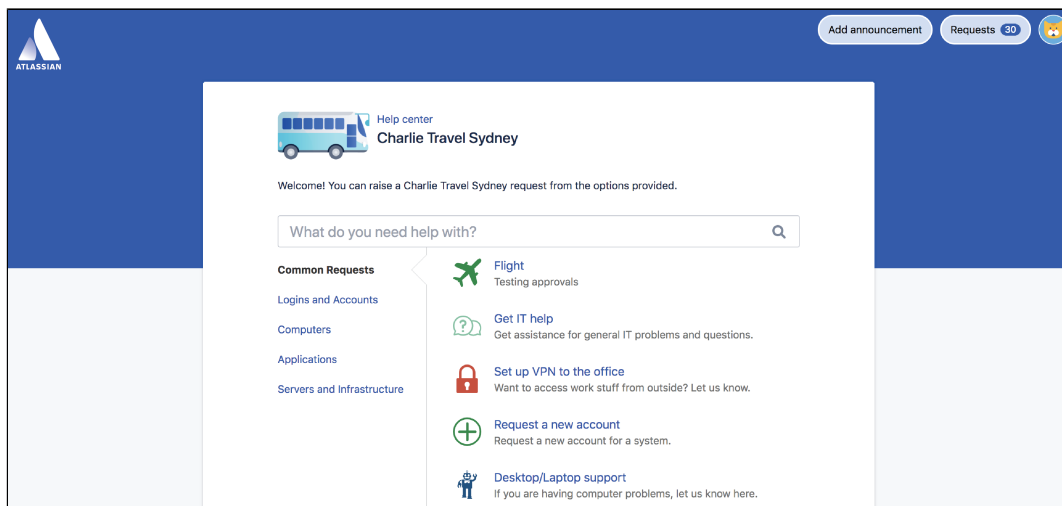
Here, you will set up request types, brand your customer portal, link your service project to an email account, and manage users:



Customer portal



This link lets you navigate the customer view of your service project:



Now that you're familiar with your service project workspace, you can set up your own Jira Service Management site and add your first project.

Let's go!

Setting up your service project

1. Setting up your service project
2. Creating service project request types
3. Making queues for your service project teams
4. Adding service project agents
5. Customize your service project channels
6. Bring your service project to the next level
7. Introduce customers to your service project
8. Explore a sample project

You'll need a working Jira Service Management instance to complete this tutorial, and we have installation instructions for both Windows and Linux operating systems.

- [Windows installation instructions](#)
- [Linux installation instructions](#)

If you have an existing Jira Service Management site, [skip ahead](#) to create a service project. If your administrator has set you up as a project admin for an existing project, jump to [Step 2](#) to create your request types.

Create a project

Jira Service Management comes with default project templates that you can use to suit your team's needs. There are three templates you can choose from:

- The **Basic Service Desk** template is set up for internal business teams, like HR, finance, or small IT teams. The template comes with just a few recommended request forms and is easy to customize and expand with your needs.
- We recommend the **IT Service Desk** template for IT teams who maintain a more complex infrastructure. The template comes with ITIL-inspired workflows for change, incident, and problem management.
- If you help external customers and want to collect bug reports or take suggestions for new feature, we recommend the **Customer service** desk template. The template comes ready for your customers to request technical or billing help, and report issues.

For now, let's get you set up with a basic service project.

1. If you're working with an existing Jira Service Management instance, select **Projects > Create Project** from the top navigation bar of your instance. If you've just installed and set up Jira Service Management, you'll have the option to Create a new project by selecting **Create new project**.
2. Select "**Basic Service Desk**" as the project type.
3. Name your project. In this example, we'll use the project name "Charlie Travel Franchises". The project key should be automatically populated, but you can change the key if you'd like. If you see options to link another application, leave these options unchecked.
4. Select **Submit** to create your project.

Nice work! You now have a service management site with one project. You will now learn to set up request types, which define the requests customers can submit to your team's service project.

[Next](#)

Creating service project request types

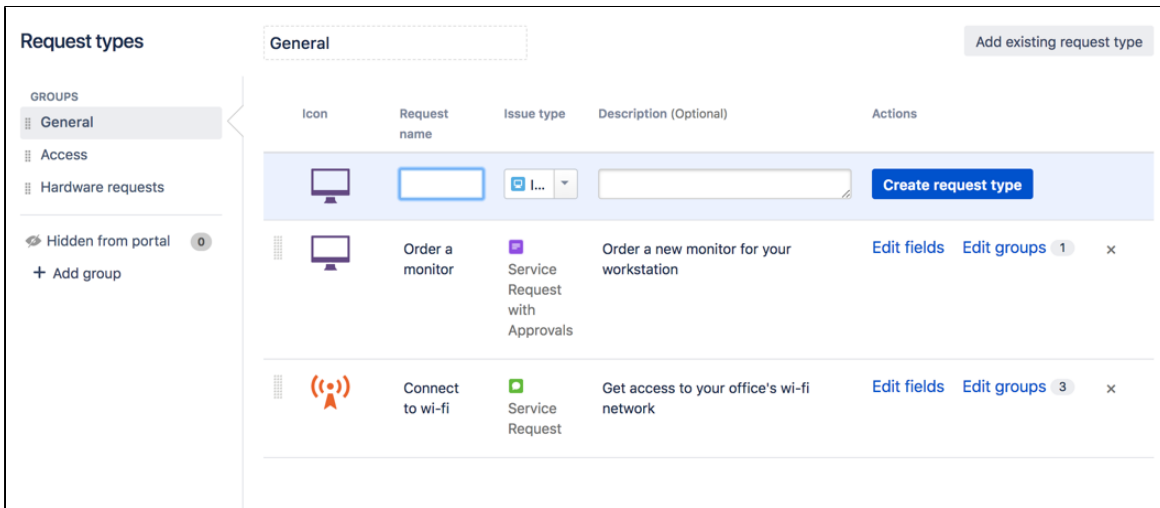
1. [Setting up your service project](#)
2. Creating service project request types
3. Making queues for your service project teams
4. Adding service project agents
5. Customize your service project channels
6. Bring your service project to the next level
7. Introduce customers to your service project
8. Explore a sample project

Request types let you define and organize incoming issues so your service project team can more efficiently help your customers. If you're moving from an existing help desk application, you can add your existing request categories during this step.

If you're setting up service project request types for the first time:

- Think about how your customer would write a request, for example 'Order a new monitor' over 'Submit a hardware request'.
- Break things down into smaller chunks, such as 'Get help with printers' or 'Get wi-fi access'.
- Avoid specialist terminology; think 'I need access' more than 'Deploy SSH key'.

By the end of this step, your project's request type page should look something like this:



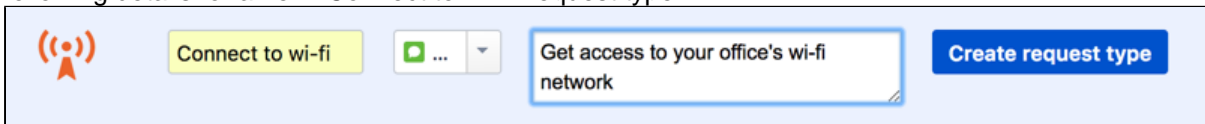
Requests vs. Issues

Remember that customers submit requests to your service project and your team picks up the corresponding issues to work on.

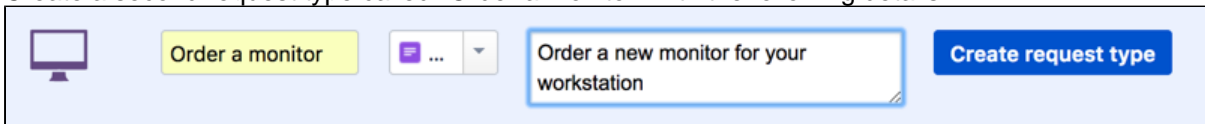
Create new request types

Let's go ahead and add two new request types, so you can familiarize yourself with the request type configuration options.

1. In your new service project, select **Project settings > Request types**.
2. In the new request type form at the top of the page, change the request type icon and enter the following details for a new "Connect to wi-fi" request type.



3. Create a second request type called "Order a monitor" with the following details:



Select **Create request type** when finished.

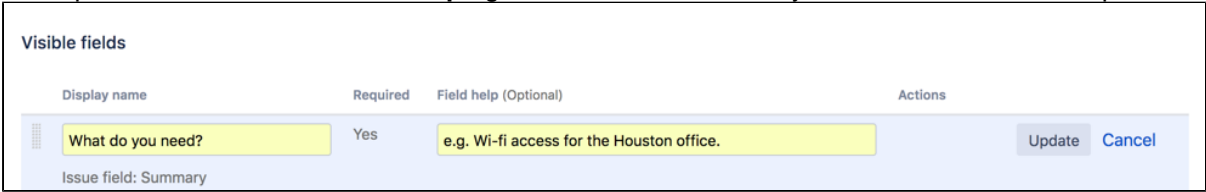
Edit the fields your customers see

Now that you have requests in the customer portal, you can prompt your customers to give you the information you need to help them quickly. These simplified fields help customers understand what information they need to provide when submitting a request.

Let's add some fields to your request types, so you can collect some additional information.

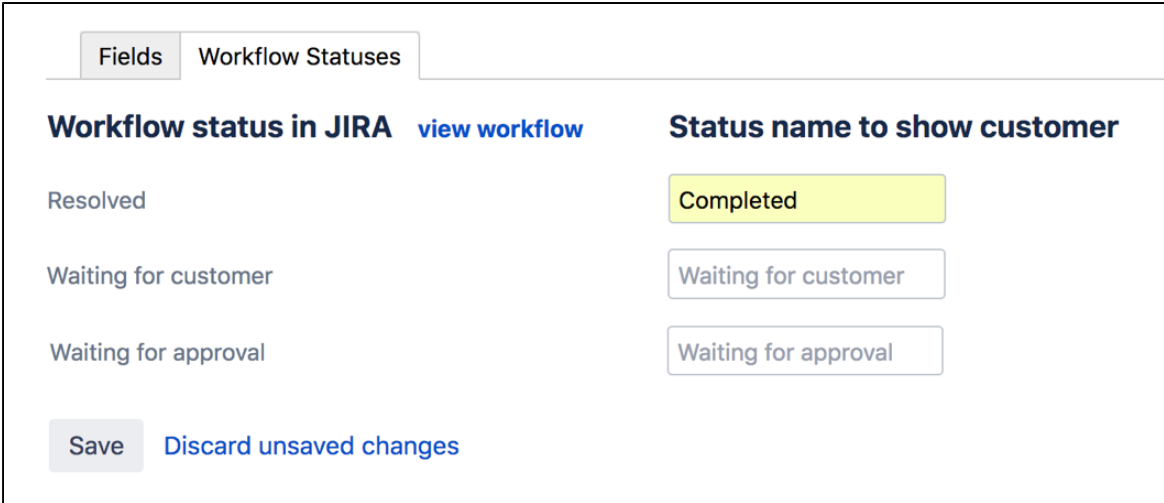
1. For the "Connect to wi-fi" request type, select **Edit fields**.

- 2. Under **Visible fields**, click the "Summary" display name and rename it to "What do you need?". Add some placeholder text to the **Field help** together useful details from your customers. For example:



Select **Update** when finished.


- 3. Select **Add a field** to add the "Priority" field to the request form and select **Apply**.
- 4. On the **Workflow Statuses** tab, you will see the default Jira workflow status names displayed on the left hand side. You can change how these statuses appear to customers by editing the *Status name to show customer* fields as shown:



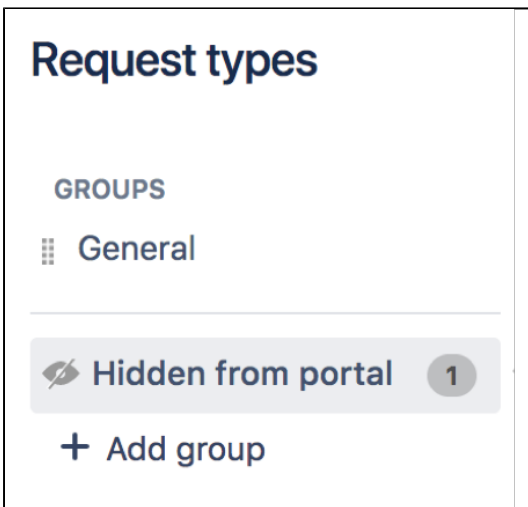
- 5. Select **View this request form** to see how your changes appear in the customer portal.

Organize your requests with groups

A group is simply a category you can assign to each request type. In the customer portal, your request types are organized vertical tabs based on your groups.

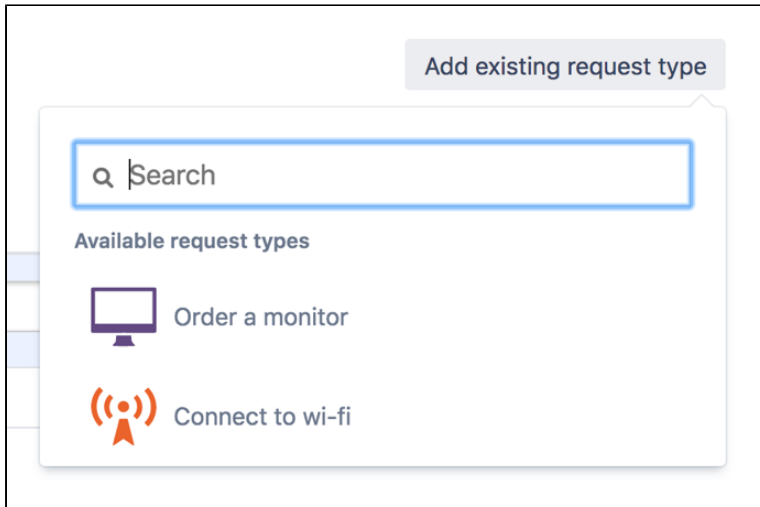
 We think groups are helpful if you have *seven or more* request types.

Go to **Project settings > Request types**. You should see your groups in the sidebar:



Let's add a few groups to help your customers find the request they need:

1. To add request groups, select **Add group**. Add two groups for your new request types, "Hardware requests" and "Access".
2. When viewing your "Access" request group, select **Add existing request type** and choose your "Connect to wi-fi" request type:



3. Switch to your "Hardware requests" group in the sidebar.
4. Select **Add existing request type** and choose your "Order a monitor" request type.
5. Open the customer portal link from your project sidebar to see your requests organized into groups.
6. To rearrange the order of how your groups appear in the customer portal, go back to your project settings and drag and drop the groups in the request type sidebar.

Create a request from the customer portal

1. Keep the customer portal preview open, so you can create test requests from a customer's perspective.
2. Select the "Connect to wi-fi" request type.
3. Enter "Test wi-fi request" in the open field and select Medium priority.
4. Click **Create** to complete your request and view the open request in the customer portal.
5. Click **Close** to exit the customer view and return to your service desk project.

Excellent work! You now have four request types and a new issue in your project. Next, you will learn how to sort these issues into queues, which will allow you to manage your team's workload.


[Next](#)

Making queues for your service project teams

1. [Setting up your service project](#)
2. [Creating service project request types](#)
3. Making queues for your service project teams
4. Adding service project agents
5. Customize your service project channels
6. Bring your service project to the next level
7. Introduce customers to your service project
8. Explore a sample project

Your teams will spend the majority of their time working out of the queues you set up. Agents do not have the permissions to add new queues or configure existing ones; however, Jira Service Management queues allow you to automatically triage and prioritize issues for them. If you want your team to focus on requests that must be completed by next week, for example, you can set up a queue that only contains requests with a set due date in that week.

Your site comes with preconfigured queues (for example, "Unassigned issues"), but let's go ahead and create three new queues for your team:

1. From your service project sidebar, select **Queues** ()
2. Select **New queue** and name your first new queue "Service requests".

- Define the issues you want to appear in this queue by selecting the following drop-down menus: **Type** (select "Service Request"); **Status** (select "Waiting for support"), and **Resolutions**(select "Unresolved"):

New queue

Name

Issues to show

More ▾ Service Request ▾ Waiting for sup... ▾ Unresolved ▾

Columns

More ▾ Key × Summary × Created × Updated × Due Date ×

Create
Cancel

- Select the following column names that will display in this queue from the **More** menu: "Key", "Summary", "Created", "Updated", "Due Date". You can reorder the columns by dragging the name (for example, "Key") across the column field.
- Select **Create** to add this queue to your team's workspace.
- Create two new queues with the following two search queries: "Completed requests" for Service requests that have been successfully resolved.

New queue

Name

Issues to show

More ▾ Service Request ▾ Resolved ▾ Resolution: All ▾

Columns

More ▾ Key × Summary × Created × Updated × Due Date ×

Create
Cancel

"Due this week" for requests that must be completed in the next week

New queue

Name

Issues to show

More ▾ Type: All ▾ Waiting for sup... ▾ Resolution: All ▾ Label: All ▾

Due Date: All ▾

Now overdue

More than minutes overdue

Due in next weeks or is overdue

Between and

In range to

Update
Close

7. Reorder your saved queues by clicking and dragging them to their new location.

You now have three new queues in your project! You will next learn how to add agents to your siteso you can get your teams up and running with Jira Service Management.

Next

Adding service project agents

1. [Setting up your service project](#)
2. [Creating service project request types](#)
3. [Making queues for your service project teams](#)
4. [Adding service project agents](#)
5. [Customize your service project channels](#)
6. [Bring your service project to the next level](#)
7. [Introduce customers to your service project](#)
8. [Explore a sample project](#)

There are two default project roles you can assign users to in Jira Service Management:

- Service Desk Customers who create requests via email or the customer portal
- Service Desk Team who view and respond to these requests

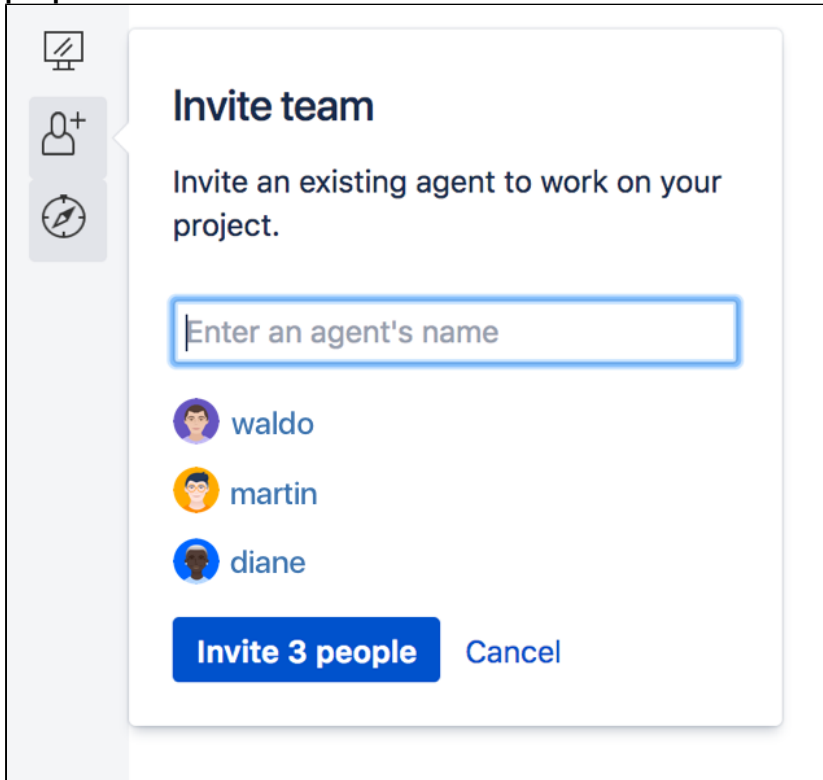
As the user who created this project, you have been automatically added to the Administrators project role.

Add your agents

Project administrators can add agents with existing user accounts to their project. If you are a project administrator, you will need to contact your site administrator to add user accounts for new agents. Make sure you're signed in as an administrator for this step, you can invite three new agents to your project - **Diane**, **Martin**, and **Waldo**:

1. In your project sidebar, select **Invite team**.

2. Enter the email addresses for your new agents and select **Invite 3 people**.



3. The agents are added to the Service Desk Team role in **Project settings > Users and roles**.

Assign issues to agents

Your agents will generally work out of queues that have issues automatically triaged into them. Let's test out manually assigning issues in case you ever come across a customer request that you want a certain agent or team to handle.

1. From the **Queues** tab, open one of your test requests by clicking the issue summary or issue key.
2. Select **Assign**.
3. Enter Waldo in the **Assignee** field and select **Assign**. When Waldo signs into Jira Service Management, this issue appears in his queue.
4. Assign another test issue to Diane.

Add your customers

You don't need to add customers to your service project during this tutorial, but let's check out where you would add them, so you're familiar with the steps:

1. From your project sidebar, select **Customers**.
2. Select **Add customers** in the top right corner and enter one or more email addresses.
3. When you select **Invite**, the customers receive an email invitation with a link to your customer portal, where they can complete the signup process.

Public customer signup

You can have your customers sign up for their own accounts (without an individual email invite) by enabling [public signup](#).

You're almost done! You have now added 3 agents to your service project and reviewed the process of assigning issues to these agents. You can now customize your customer portal and share it with the rest of your team.

[Next](#)

Customize your service project channels

1. [Setting up your service project](#)
2. [Creating service project request types](#)
3. [Making queues for your service project teams](#)
4. [Adding service project agents](#)
5. [Customize your service project channels](#)
6. [Bring your service project to the next level](#)
7. [Introduce customers to your service project](#)
8. [Explore a sample project](#)

Service project customers can send you requests in two ways:

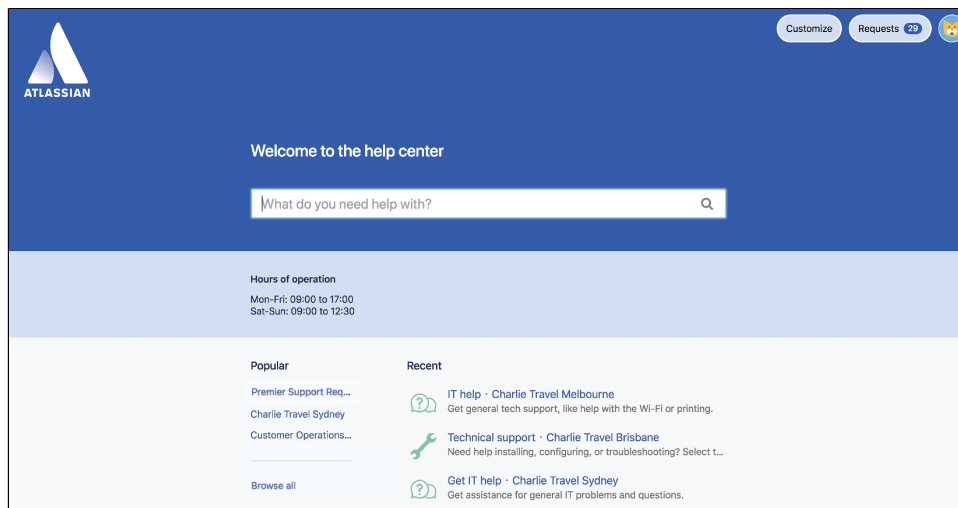
- through an online portal
- to an email account you've linked to your service project.

Let's customize the portal and add an email account, so customers can get in touch.

Customize the help center

Your service project comes with an online help center for your customers. From the help center, customers can browse all the service projects they have access to.

It will look something like this:



Jira admins, project admins, and agents with permission can brand the help center in a number of ways. Here's how:

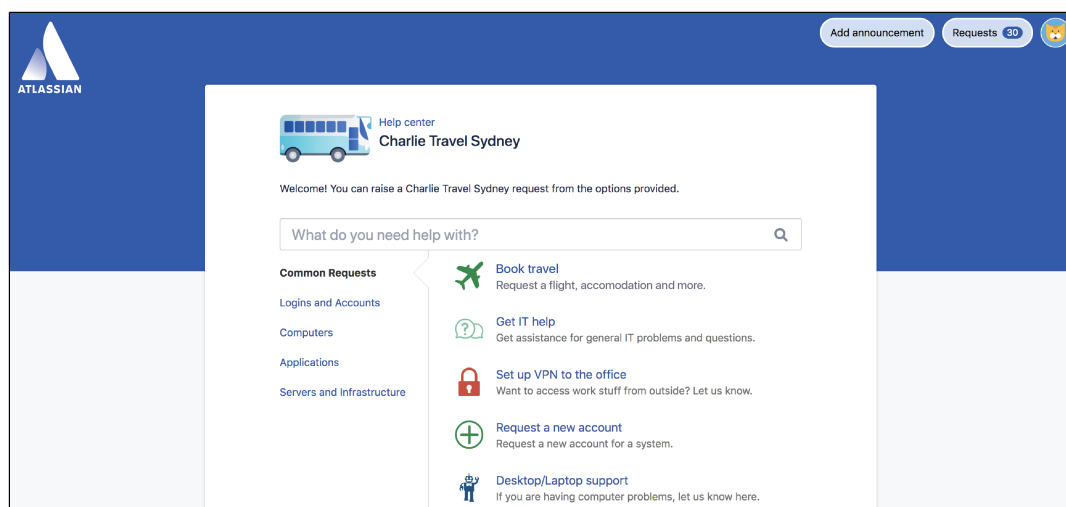
1. Go to **Jira administration** > **Applications** > **Jira Service Management** > **Configuration**.
2. Below **Help center**, click the link **View and customize**.
3. In the customize panel on the right hand side, do any the following:
 - Add a help center announcement
 - Add an announcement to the login screen
 - Give your help center a name
 - Add a logo
 - Add a banner image or banner background color
 - Change the color of some components
4. Choose **Save changes** to publish your announcement.

To learn more about the help center check out [Using the help center](#).

Customize the customer portal

Now that your help center looks inviting, you can customize the customer portal for each of your service desks. Add a name and logo to your customer portal so that people know they're sending requests to the right team. The banner, links, and button colors are all inherited from the help center.

It will look something like this:



To customize your customer portal:

1. In your service project, go to **Project settings** > **Portal settings**.

2. Edit your customer portal name and intro text by typing in the fields.
3. Save any edits by clicking the tick icon in the field you're editing.
4. Add a customer portal logo by selecting **Use a custom logo**.
5. Select **Save logo**, once you've uploaded your image.

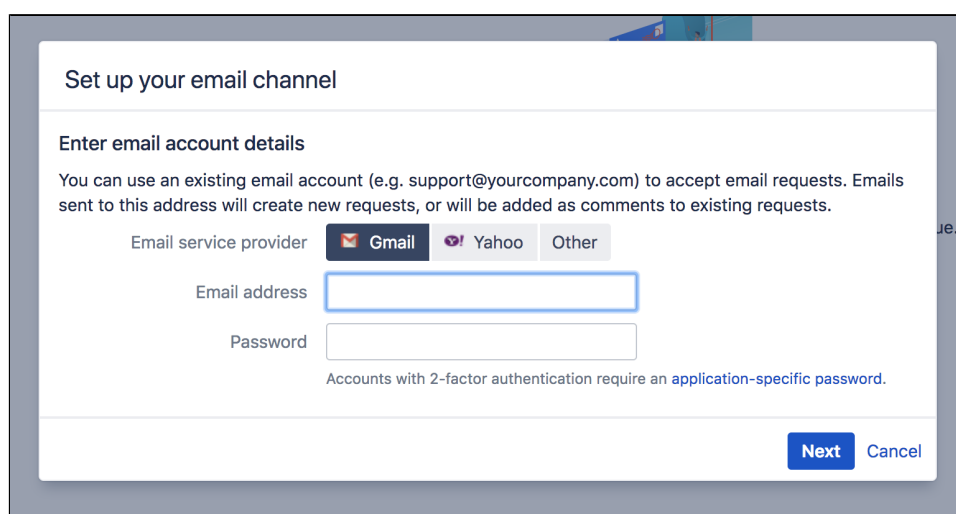
To learn more about the customer portal check out [Configuring the customer portal](#).

Link an email account

Customers can also create requests and communicate with your team by email. If you haven't already, create an email address that you'd like your team to be contacted on.

To link an email to your service desk project:

1. Go to **Project settings > Email requests**.
2. Select **Add email address** and fill in the details. If you use 2-step-verification for Gmail, generate an [application-specific password](#) when adding your email account details.



3. Once you've linked an email to your service project, we'll send a test email to your inbox and create a test request in your service project.

To learn more linking an email to your service project, check out [Receiving requests by email](#).

- ✔ If you use POP, make sure the email account you choose for this channel has an *empty inbox* so you do not lose any existing emails.

Tell customers about your service project

Now that your service project is ready to receive requests, share the project's email address (for example, helpdesk@example.com) with your customers or give them a direct link to the customer portal.

You can give either, or both, of the following URLs to your customers, depending on your setup:

- URL to a customer portal give this URL to your customers if you've enabled public signup and want them to sign up for accounts on their own.
- URL to the help center here they will see all the service projects they have access to. The URL is: `http://<computer_name_or_IP_address>:<HTTP_port_number>/Jira/servicedesk/customer/portals`

✔ Here are other ways that some customers promote their service project:

- post a link on the intranet or other communication channels
- email their customers to tell them there's a new way to get help
- put posters up in communal areas around the office.

You've now finished setting up your service project! Continue to learn more advanced tips on tracking your team's progress and supporting your customers.

[Next](#)

Bring your service project to the next level

1. [Setting up your service project](#)
2. [Creating service project request types](#)
3. [Making queues for your service project teams](#)
4. [Adding service project agents](#)
5. [Customize your service project channels](#)
6. [Bring your service project to the next level](#)
7. [Introduce customers to your service project](#)
8. [Explore a sample project](#)

Now that you have your basic service project up and running, you can learn about the following advanced features:

- [Serve your customers and your team better with SLAs](#)
- [Track your team's success with reports](#)
- [Solve requests faster with a knowledge base](#)

Serve your customers and your team better with SLAs

Service-level agreements (SLAs) help you communicate service agreements to your customers and keep track of your team's performance. An SLA consists of a time metric and a corresponding goal or target.

As the administrator, you can configure each SLA metric and goal using the Jira Service Management SLA designer. SLA information will appear in the internal issue and your agents can also view SLA goals by going to **Reports > Workload** when they log in to your service desk project.

Let's have a quick look at where you can create a new SLA metric:

1. In your service project, select **Project settings > SLAs**.
2. Select **Create SLA** to create a new SLA metric for your service desk project.
3. For more information, check out [Setting up service level agreements \(SLAs\)](#).

Track your team's success with reports

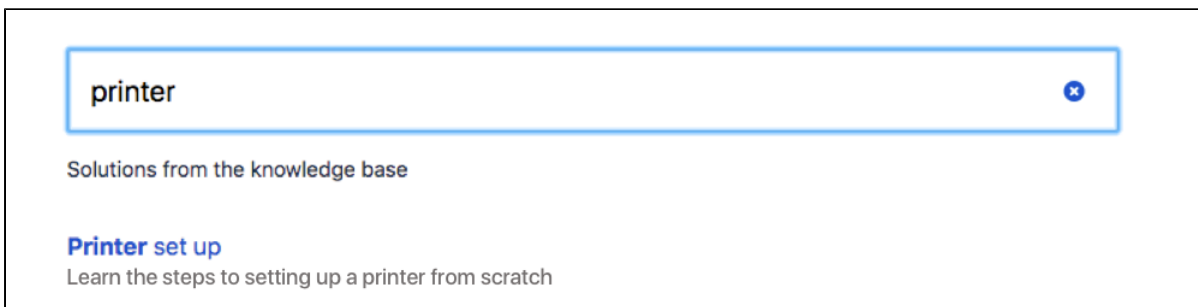
Jira Service Management lets you display selected SLA metrics and goals in interactive reports. Reports can be used to help you visualize your team's performance so you can identify bottlenecks and optimize your team's workload. Your team of agents can then view the read-only versions of your reports to see how they are tracking towards their goals.

Let's now have a quick look at the **Reports** tab:

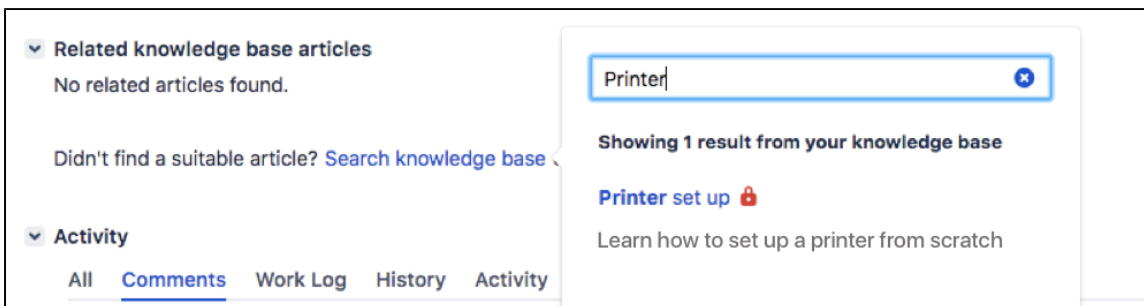
1. From your service project sidebar, select **Reports** to view the pre-configured reports in your project.
2. Select **+New Report** to create a new report, or simply edit one of the pre-configured reports.
3. For more information, check out [Setting up service project reports](#).

Solve requests faster with a knowledge base

You can deflect common requests and solve issues faster by linking a Confluence knowledge base to your service project. When you link a knowledge base to your project, customers can search for solutions in the portal and help center before they raise a request:



A knowledge base helps agents, too. When they work on issues, they can see related knowledge, search for solutions, and create new articles from issues:



Customers can use the articles to self-service problems, and agents can use them to solve requests faster. Everybody wins.

To learn more about linking a knowledge base to your service project, see [Set up a knowledge base for self-service](#).

You're almost done! We'll now review the ways customers can contact your team and be informed of updates to their requests.

[Next](#)

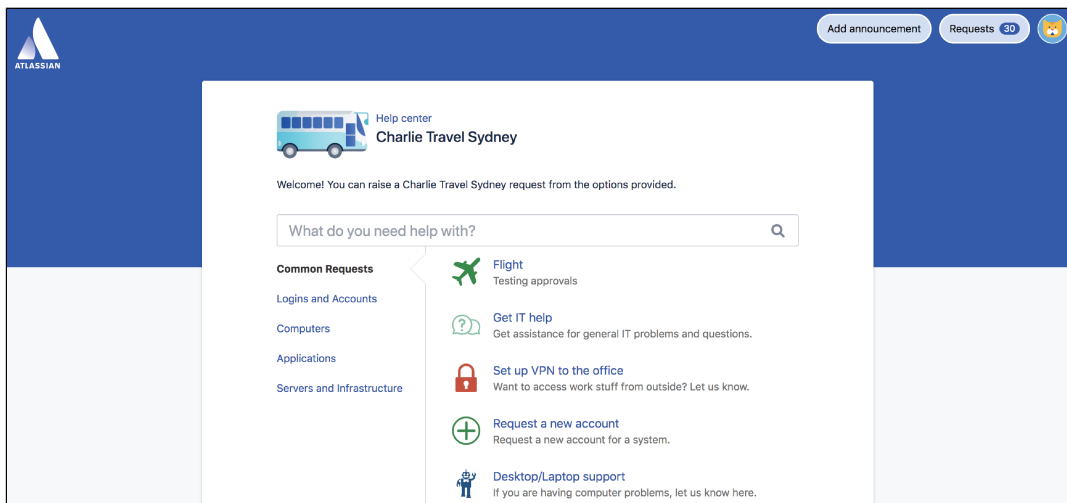
Introduce customers to your service project

1. Setting up your service project
2. Creating service project request types
3. Making queues for your service project teams
4. Adding service project agents
5. Customize your service project channels
6. Bring your service project to the next level
7. Introduce customers to your service project
8. Explore a sample project

Now that you have set up your project in a way that serves both your agents and your customers, it's time to show your customers how to start using Jira Service Management.

Create requests through the customer portal

1. Visit the customer portal.
2. Pick an option that matches what you need and fill in the details of the request.



Create requests by email

Another way of creating requests is by sending emails to a linked service project. Ask your service team if they are set up to receive email in their service project. If they are, simply email them a request directly and keep the conversation going from your inbox.

Create requests in multiple service projects

To send the same request to multiple teams, you have the following options:

- If all of the teams you want to contact have linked their service project to an email account, you can easily create the request by sending one email message to all linked service project email accounts.
- If the teams you want to contact have not all linked their service project to an email account, you will need to create the request in each service project one by one, either through their customer portal or sending emails.

Track and comment on requests

Use the customer portal to see all requests you have created, read comments from agents as they are updated, and check the status of a request. You can add comments and attachments to requests on the customer portal as well.

Another way of tracking requests is through email notifications. You receive email notifications when agents respond to your requests and when the request has been resolved. To add comments to requests, you can simply reply to the email notifications and your reply will be added as a comment to the request.

Congrats! You've completed the Getting started for service project admins tutorial.

Want to learn more? Check out the home of Jira Service Management documentation [here](#).

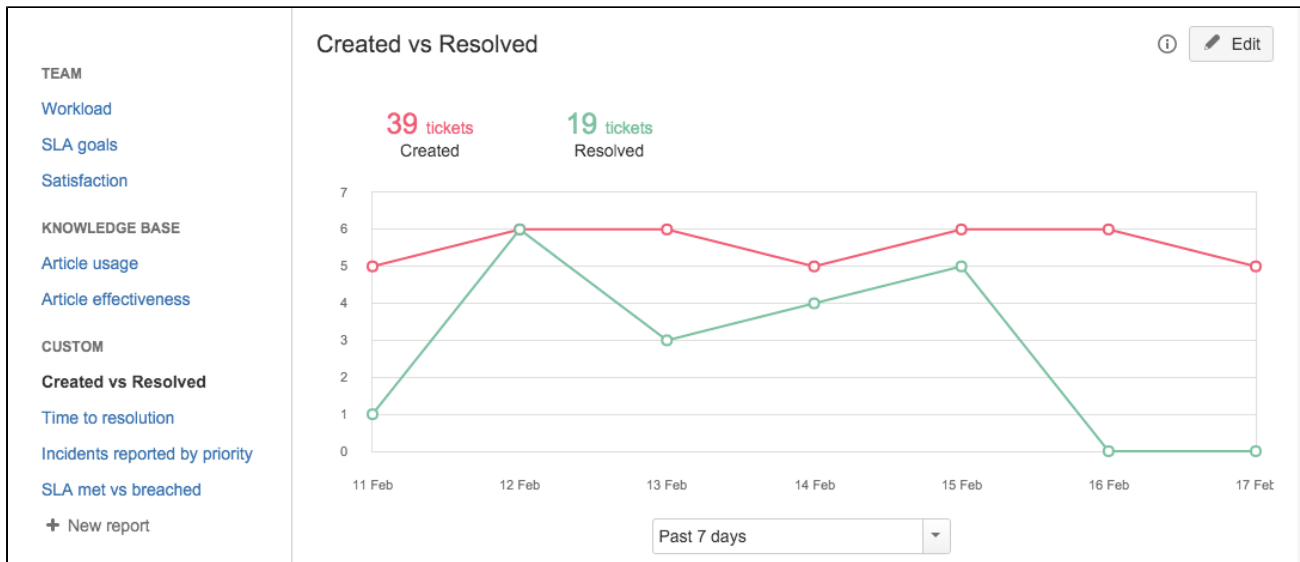
Explore a sample project

Jira Service Management comes with sample data to help you explore and learn how to use key features.

When you create a sample project, it gets populated with issues that new team members can use to learn about concepts like queues, SLAs, and generate reports like the one below without fear of affecting any real work.

On this page:

- [Create a sample project](#)
- [Learn about key features](#)

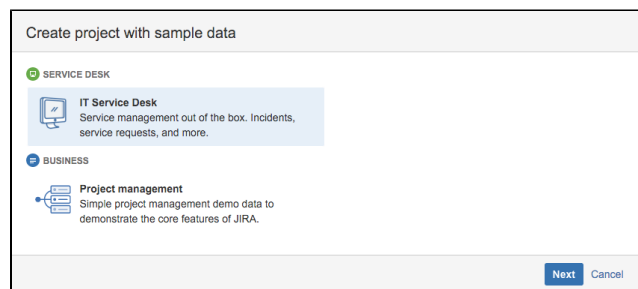


If you are a Jira Service Management administrator, we suggest you create and explore a sample project to help you and any new team members explore how a service desk project works.

Create a sample project

You need to be a Jira Service Management administrator.


1. Go to **Project > Create project**.
2. In the Create project screen, click **Create sample data**.
3. In the Create project with sample data screen, select **IT Service Desk** and click **Next**.
4. Enter a name for the sample project.
Tip: If you are creating the project for a specific user, name the project 'Sample - [name user]'. This will make it easier to find and delete later.
5. Click **Submit**.



Sample project access Depending on how user access is set up in your Jira Service Management version, you may need to [give new users access to the project](#) as well.

Learn about key features

Here are a list of tasks that we highly recommend you have a go at:



Get on board your new service desk

- ✓ Be awesome
- ✓ Learn the key concepts of Jira Service Desk
- ✓ Create a request in the customer portal
- ✓ View issues in queues, where customer requests live
- ✓ See what you can do as a service desk admin

Check out tutorials and best practices in the [documentation](#)

- Explore the customer portal and see what your customers see
- Create a new request and assign it to yourself
- View the queue, edit an issue description, or add a label
- Comment on an issue
- Try out your email channel
- Play with reports

i Finished playing with your sample project?

When a sample project has served its purpose, delete it from the project directory. You need to be a Jira Service Management administrator to do this.

Getting started for service project agents

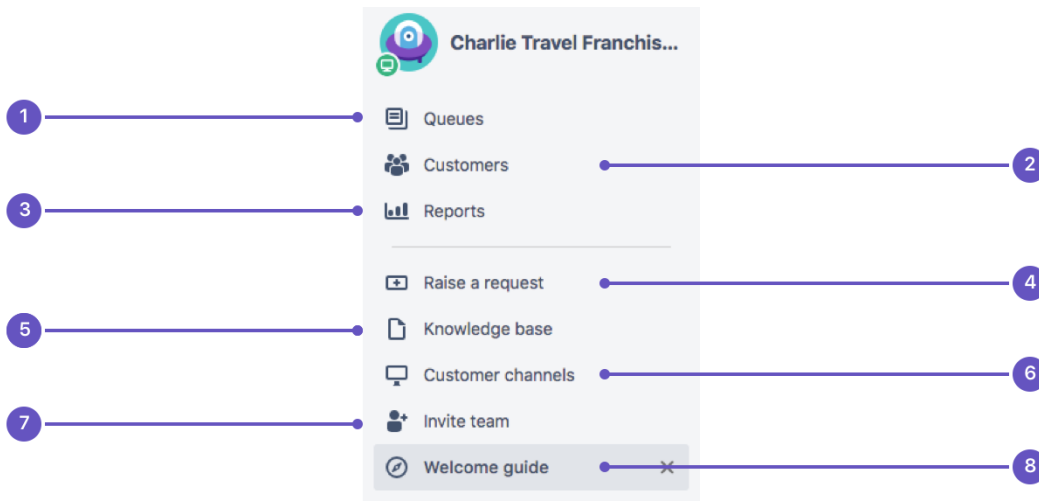
On this page, we introduce you to your workspace and walk you through the process of responding to your customers' requests.

On this page

- [Navigate your workspace](#)
- [Work on customer issues](#)
- [View and create knowledge](#)

Navigate your workspace

Use the Jira Service Management sidebar to navigate your workspace:



1. **Queues:** view issues that customers have submitted to your service desk.
2. **Customers:** view your customers and requests, and invite new customers to the service project.
3. **Reports:** view reports about your team's SLA goals, knowledge base and workload.
4. **Raise a request:** raise a request on behalf of a customer.
5. **Knowledge base:** search your team's knowledge base or create an article.
6. **Customer channels:** view your service project's customer portal and email address.
7. **Invite team:** invite an existing agent to work on your project.

Work on customer issues

When customers submit requests to your service project, the issues are grouped into queues on the **Queues** page.

Select an issue's **Summary** or **Key** to view more information about an issue, or start working on an issue. The issue view looks like this:

The screenshot shows a Jira issue titled "The printer is not working" (ID: CTS-13). The issue is currently in the "Waiting for support" status. The interface includes a top navigation bar with options like "Edit", "Comment", "Assign", and "More". Below this, there are sections for "Details" (Type: Service Request, Status: WAITING FOR SUPP...), "Description", "Attachments" (with a "Drop files to attach, or browse." prompt), and "Activity" (with tabs for "All", "Comments", "Work Log", "History", "Activity", "Transitions"). On the right side, there are sections for "SLAs" (Time to first response within 4h, Time to resolution within 1d), "People" (Assignee: Unassigned, Reporter: Alana Grant), and "Service Desk request" (Request type: Get IT help, Customer status: Waiting for support, Channel: Unknown). A "Return to queue" link is visible in the top right corner.

From here, you can work with customers and your team to resolve requests. If your service project is linked to a knowledge base, you can also view, share, and create knowledge from the issue.

Comment on an issue

When you select **Respond to customer**, add an attachment, or comment on an issue, you can share your response with the customer, or comment internally.

- If you share with the customer, the customer is notified and can see the comment or attachment when they view the request in the customer portal.
- If you make an internal comment, the customer isn't notified and can't view the comment or attachment on the customer portal. People who are watching the issue are notified, and your team can see the comment when they view the issue in your service project.


React to a comment

 You need a Data Center license to use comment reactions.

Add a comment reaction

You can use comment reactions in the issue view instead of replying to a comment.

To add a reaction to a comment:

1. Go to your issues comment section.
2. Select the **add emoji icon** .
3. Select the relevant reaction.

The screenshot shows a comment from "Friendly Robot" added on 19/Nov/21 at 2:57 PM. The comment text is "Hey, it's Friendly Robot here. I'll be handling your request today!". Below the comment, there is an "Add reaction" menu with several emoji options: thumbs up, thumbs down, fire, smile with hearts, sad face, and crying face.

Enable or disable comment reactions

Your customers can react to comments with emojis, too. As the customer portal comment reactions are disabled by default, you first need to allow their use.


To enable comment reactions:

1. Go to **Project settings > Portal settings**.
2. Go to the **Comment reactions** section.
3. Select **Allow comment reactions on the customer portal for this project**.

You can fully disable comment reactions via the global system settings.

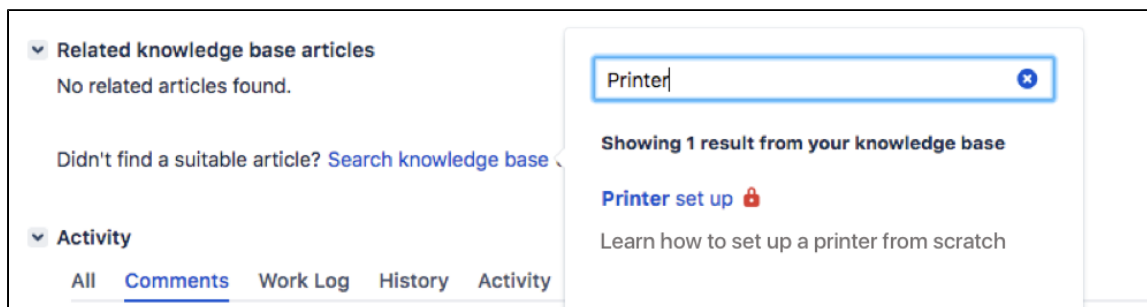
Collaborate on an issue

You might want to share the issue with other customers who have a similar problem, or with other people on your team who can help you resolve the issue. Here are some ways you can involve other people in the issue:

- Select **Share**  to email a link to the issue to other people on your team.
- Use @mentions (@username) to mention a team member in a comment.
- Add watchers to involve other users from your Jira site. Someone leaves an internal comment or transitions an issue.
- Add request participants to share the issue with other customers or organizations. Request participants receive the same notifications as the reporter.

View and create knowledge

If your service project has a linked knowledge base, you can use knowledge articles to solve issues faster:



For example, if you get a lot of requests about printing problems, you can write a step-by-step article that helps customers troubleshoot the problem themselves:

- When customers search for help from the customer portal, they can use articles to solve problems instead of raising requests.
- When you work on an issue, you can share the article with customers so they can try to troubleshoot on their own.

You can also write internal articles and reference them while you work on an issue. Articles that the reporter can't view are marked with a red padlock.

When you create a new article from an issue, you can choose a handy *how-to* or *troubleshooting* template to guide you. The issue summary and description become the article's title and body text.

Nice work! Want to learn more? Proceed to [Working on service projects](#) to learn more about what a service project agent can do.

Administering service projects

Welcome to the source of truth for Jira Service Management administrator knowledge and power.

This section is for Jira Service Management project administrators. You're in the right place if you're the one who adds agents and sets up your service project. If you're new to Jira Service Management, check out our [getting started guides](#).

Get started

Here are some pages to get you started:

- [Managing access to your service project](#)
- [Configuring the customer portal](#)
- [Receiving requests by email](#)
- [Setting up service project users](#)
- [Setting up queues for your team](#)
- [Automating your service project](#)
- [Managing service project notifications](#)
- [Setting up request types](#)
- [Setting up SLAs](#)
- [Setting up approvals](#)
- [Setting up service project reports](#)
- [Default service project configuration](#)
- [Using Jira applications with Confluence](#)

Managing access to your service project


Go to **Project settings** > **Customer permissions** to choose who can raise requests in your service desk and who your customers can share requests with.

On this page:

- [Choose who can raise requests](#)
- [Choose who customers can share requests with](#)
- [Choose whether customers can vote for requests](#)
- [Which settings are best for my team?](#)
- [Security advice: Choosing who can raise requests](#)

Choose who can raise requests

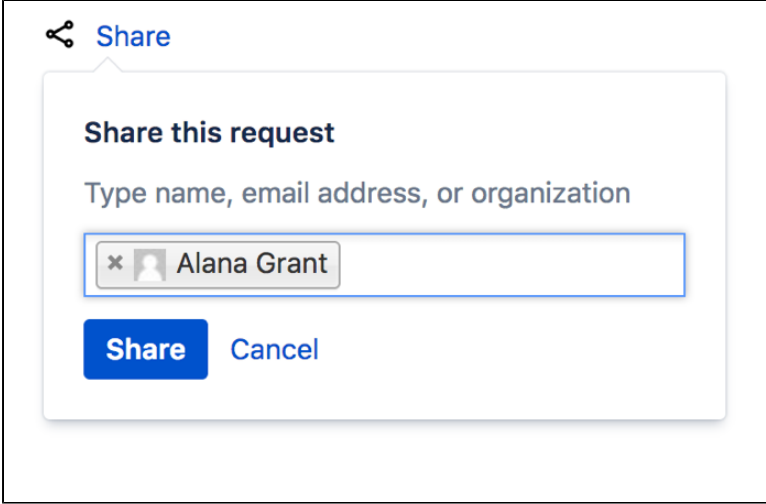


People need to be customers to raise requests in your service desk. You can let your team control who becomes a customer, or let customers create their own accounts in your service desk.

Who can raise requests	Description
Customers who are added to the project	Your team adds customers to the project via the Customers page, or by raising requests on their behalf.
Customers who have an account on this Jira site	People with accounts on your Jira site are automatically added to the Customers list and can raise requests.
Anyone can email the service project or raise a request in the portal	<ul style="list-style-type: none">• New customers can create their own accounts in your service desk via the customer portal.• Raising an email request automatically creates an account for the sender.• If you allow customers to share requests, then people they share with also become customers and can raise requests.• A honeypot technique is enabled to help prevent spambots from creating accounts through the customer portal.• If login-free portal has been enabled by your Jira administrator, and this option is selected, anyone will be able to access your portal and raise requests without logging in. Learn more about login-free portal <p>If this option is disabled, then your Jira administrator has not turned on public signup for service desks on this Jira site. Learn more about enabling public signup</p> <div style="border: 1px solid #ccc; padding: 10px;"><p> When you choose this option, you have no control over who can create an account and raise requests. We strongly recommend that you use the <i>Customers who have an account on this Jira site</i> option instead, unless the requirement for a Jira account won't work for your use case.</p><p>Read more in our security advice</p></div>

Choose who customers can share requests with


You can allow customers to share requests with their organizations, anyone in the service desk, or people who aren't customers yet. The people customers share with become participants in the request. Request participants can comment on and share requests, and receive the same notifications from Jira Service Desk as the reporter. [Learn more about request participants](#)

The following table describes the ways customers can share requests:

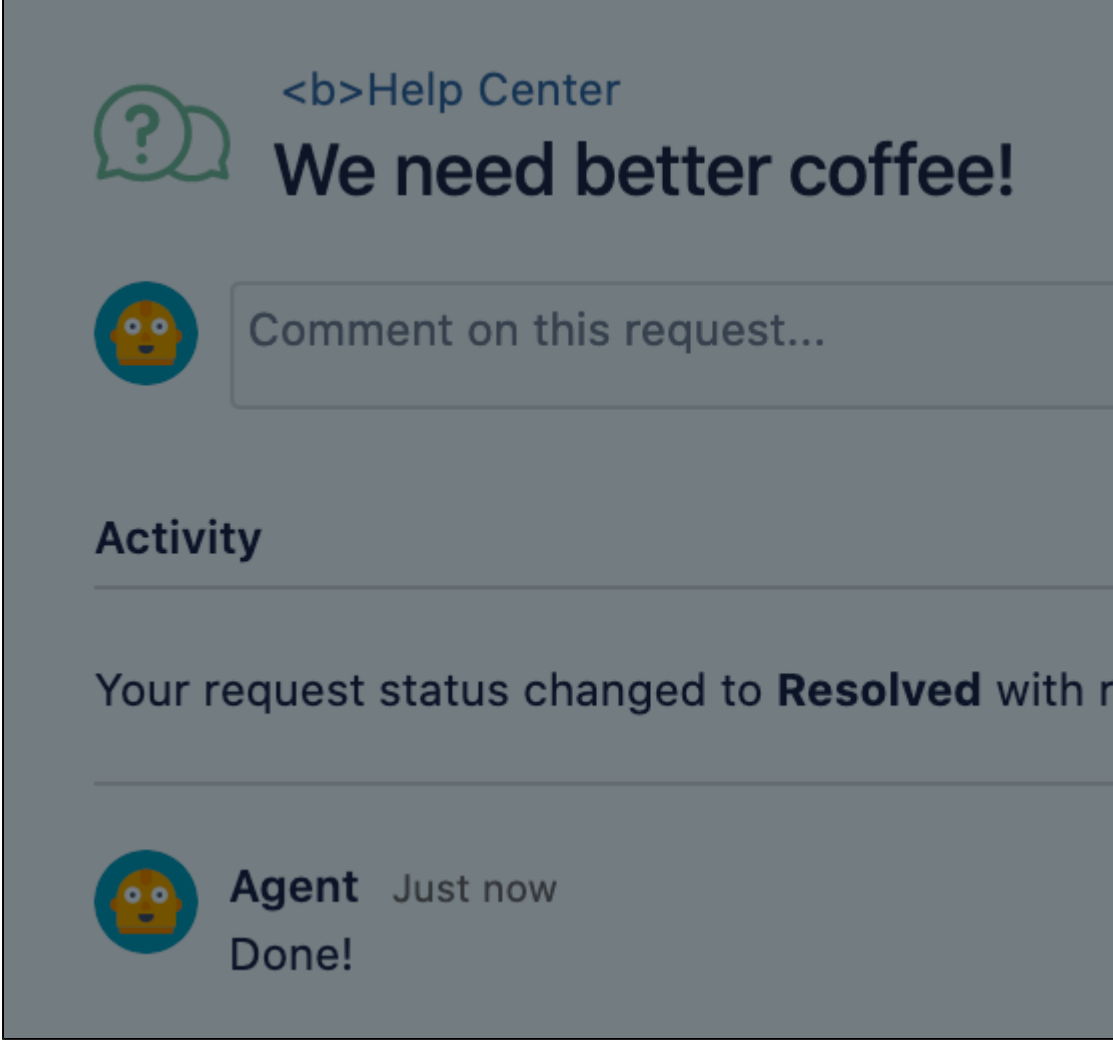
Who customers can share with	Description
Other customers in their organization	<ul style="list-style-type: none"> Customers can share requests with their organization, or raise a private request. Customers can search their organization for people to share with: <div data-bbox="392 472 1161 974" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">  <p>Share this request</p> <p>Type name, email address, or organization</p> <p>×  Alana Grant</p> <p>Share Cancel</p> </div> <ul style="list-style-type: none"> Customers who aren't in an organization can't share requests.
Any customer, by typing an email address	<p>Customers can share requests with other customers in their organization (like in the option above), and also:</p> <ul style="list-style-type: none"> Share requests with anyone in the service project, if they know their email address If anyone can email the service project or raise a request in the portal, then customers can also share requests with people who aren't customers yet. They can provide email addresses of these people and Jira will create accounts for them. Such unregistered email addresses can only be entered in the Share this request field but not a regular user picker field.
Any customer or organization, by searching in this project	<ul style="list-style-type: none"> Customers can share their requests with anyone in the project. They can also search the service project for people to share with. If anyone can email the service project or raise a request in the portal, then customers can also share requests with people who aren't customers yet. They can provide email addresses of these people and Jira will create accounts for them. Such unregistered email addresses can only be entered in the Share this request field but not a regular user picker field. <div data-bbox="347 1682 1430 1787" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> If your service project uses user picker custom fields, such as the Approvers field, choose this setting to make sure customers can select users.</p> </div>

Choose whether customers can vote for requests

You can allow customers to vote for requests directly in the customer portal. This lets them cast their votes for requests they'd like to see done without having user accounts on your Jira instance.

 To enable voting in the portal, you need to have global voting enabled for your Jira instance.

The following table describes different options for this setting:

Can customers vote in the portal?	Description
<p>Yes, any customer with portal access can vote directly in the portal</p>	<p>Customers will see an extra option to vote when they view a request in the portal. They can the</p> 
<p>No, customers can't vote directly in the portal</p>	<p>If the global voting is enabled, users with accounts on your Jira instance will still be able to vote your agents.</p>

Which settings are best for my team?

Not sure how to set up permissions for your team? Here are some suggestions for how to make customer permissions work for you:

If you're like this	Who can raise requests	Who customers can share with	Can customers vote?
You have a service project that handles contractors' leave requests. Only contractors can use the service project, and you don't want non-contractors to get confused about where to request leave.	Customers who are added to the project	Other customers in their organization	No, as contractors probably don't need to vote for their leave requests.
Your company has an IT service desk, and you want all employees to be able to create their own accounts and email requests.	Customers who have an account on this Jira site	Any customer or organization, by searching in this project	No, if your team is mostly handling requests for individual employees, such as "I need a new monitor". Yes, if your team is handling requests for general improvements. For example, employees could vote for what they'd like to be added to their office.
Your team provides software support for individuals. For example, if your company makes a free SAAS application that individuals use to manage finances, you can let your customers email bugs and questions to your service project email channel.	Anyone can email the service project or raise a request in the portal	Any customer, by typing an email address	Yes, it might be useful for individuals to vote for bugs or requests they'd like to see fixed first.

Security advice: Choosing who can raise requests

This is security advice regarding the option Who can raise requests.

When allowing **anyone to create and raise requests**, we strongly recommend that you use a separate domain for your incoming email account and configure this account to filter against automatic emails by blocking the addresses with a local-part, such as no-reply, support, servicedesk, and so on.

Reason

Configuring your Jira Service Management instance using an incoming email address with a domain at your company level can result in bad actors abusing the system to gain access to restricted services.

Example

For example, assume that your domain is `http://mycompany.com`, and you have set up a customer portal with an incoming email `support@mycompany.com`, allowing anyone to create and raise requests. Now, let's say a bad actor is trying to gain access to `teamsinspace.com` for which your company already has an account (e.g. `admin@mycompany.com`).

If the bad actor creates a new ticket in your customer portal and adds `no-reply@teamsinspace.com` as a participant, they can now go to `teamsinspace.com` and open an account for `support@mycompany.com` and provide a team name including the ticket number. Since `teamsinspace.com` includes the team name in their confirmation emails (and other websites might include other details), such an email will be sent to your incoming email address in Jira Service Management, and added as a comment to the ticket. By clicking on the confirmation email, the bad actor could open an account on `teamsinspace.com` on behalf of your company.

Mitigating this issue

Using a separate domain or sub-domain dedicated for your customer portal will help prevent this as long as you make sure to not sign up to services using that domain. Whats more, adding filters against automatic emails (e.g. no-reply) at the incoming email account level will further help mitigate the issue because no-reply emails will be blocked.

Configuring the customer portal

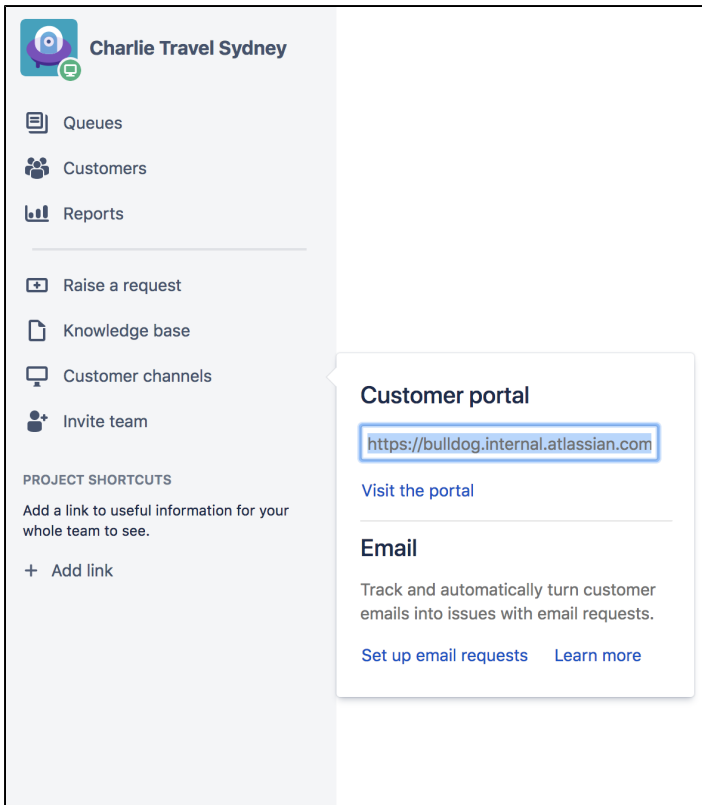
Every service project comes with an online portal that your customers use to interact with your service team.

If you link your project to a Confluence knowledge base space, customers can troubleshoot by searching for relevant articles.

To access your customer portal, select **Customer channels** in the project sidebar, then the link **Visit the portal**.

On this page:

- [Brand your customer portal](#)
- [Set up request types](#)
- [Add transitions](#)
- [Manage access to your portal](#)
- [Choose whether you need search](#)
- [Enable comment reactions](#)
- [View all customer portals in your help center](#)
 - [Share the help center](#)



i You need project admin permissions to make the changes we'll be recommending on this page.

Brand your customer portal

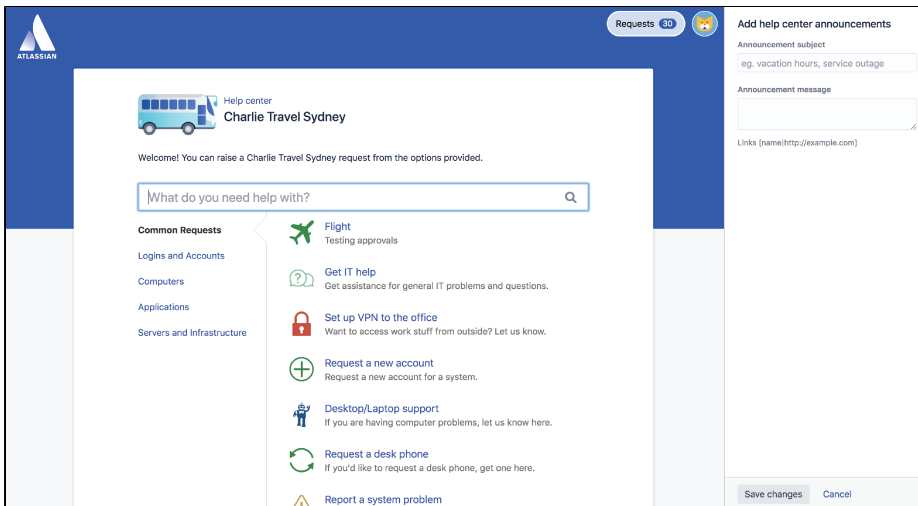
As a project admin, you can change the following components on the customer portal:

- name
- welcome message
- logo
- announcement.

The banner, links, and button colors are all inherited from the help center. Learn more at [Using the help center](#).

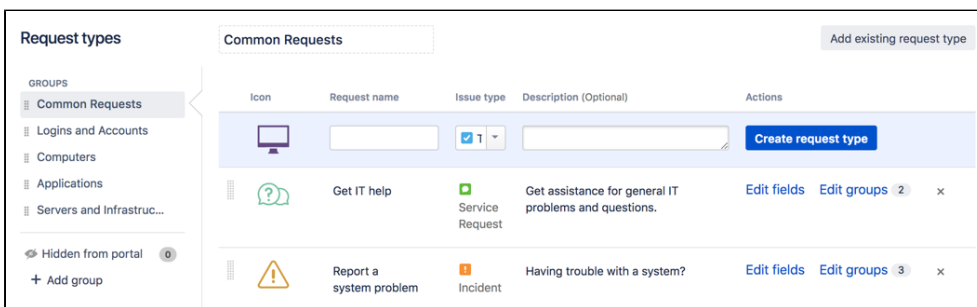
To brand the customer portal:

1. Go to **Project settings > Portal settings** and change the messaging and logo. Your logo will appear in notifications sent by your service project.
2. Select the link **View and change** under the **Announcement** heading, to add important information (for example, working hours).



Set up request types

You can customize the types of requests that customers raise from the portal. To create and manage request types, visit **Project settings > Request types**.



Jira Service Management includes several request types that address common IT help scenarios. The request types are organized into groups to help customers find what they need. For example, you can add a 'Common requests' group to help customers address issues like IT support.


To learn more about customizing request types, check out [Setting up request types](#).

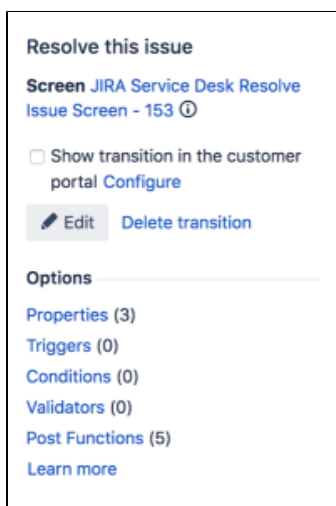
Add transitions

You can show transitions on the customer portal so that customers can change the status of their request.

For example, say an agent shares a knowledge base article with a customer and the article solves their problem. The customer can mark their request as resolved, directly from the customer portal.

To add a transition to the portal:

1. In your service project, go to **Project settings > Workflows**.
2. Select the pencil icon  beside the workflow that has the transition you want to add to the portal.
3. Select **Diagram** to open the diagram view.
4. Select the transition in the workflow, and check the box beside **Show transition on the customer portal**.



Customer transitions behave slightly differently than other workflow transitions:

- Screens don't display on the customer portal. When you add a transition to the portal, you can set a resolution for requests that customers transition.
- When an issue is transitioned from the portal, it bypasses any validators that are defined for the transition.

To learn more about workflows and transitions check out [Advanced workflow configuration](#).



If it seems like the portal transition isn't working, check if there's an automation rule in conflict with the transition. To learn more about automation rules, check out [Automating your service project](#).

Manage access to your portal

You can allow anyone to sign up for an account to your portal, or only allow customers you've added.

If a Jira administrator has configured the login-free portal setting across your Jira Service Management instance, this may mean anyone can access your portal and raise requests without logging in. To learn more about different access options, check out [Managing access to your service project](#).



The customer portal integrates with [Atlassian Crowd](#), Atlassian's single sign-on (SSO) framework. For information about integrating with third-party SSO providers check out [Single sign-on integration with Jira and Confluence](#).

Choose whether you need search


If you don't have a Confluence knowledge base linked to your project, it might be a good idea to disable the search bar on your customer portal, just to not confuse your customers. This option is often used when you want to give your portals more simplicity and space.

To disable the portal search:

1. Open your project.
2. Go to **Project settings > Portal settings**.
3. In the **Search** section, disable the search bar.

Good to know

- We'll only disable search for your specific portal, but we'll keep it on the help center.

 You need a Data Center license to use the search bar.

Enable comment reactions

Your customers can react to comments with emojis. As the customer portal comment reactions are disabled by default, you first need to allow their use.

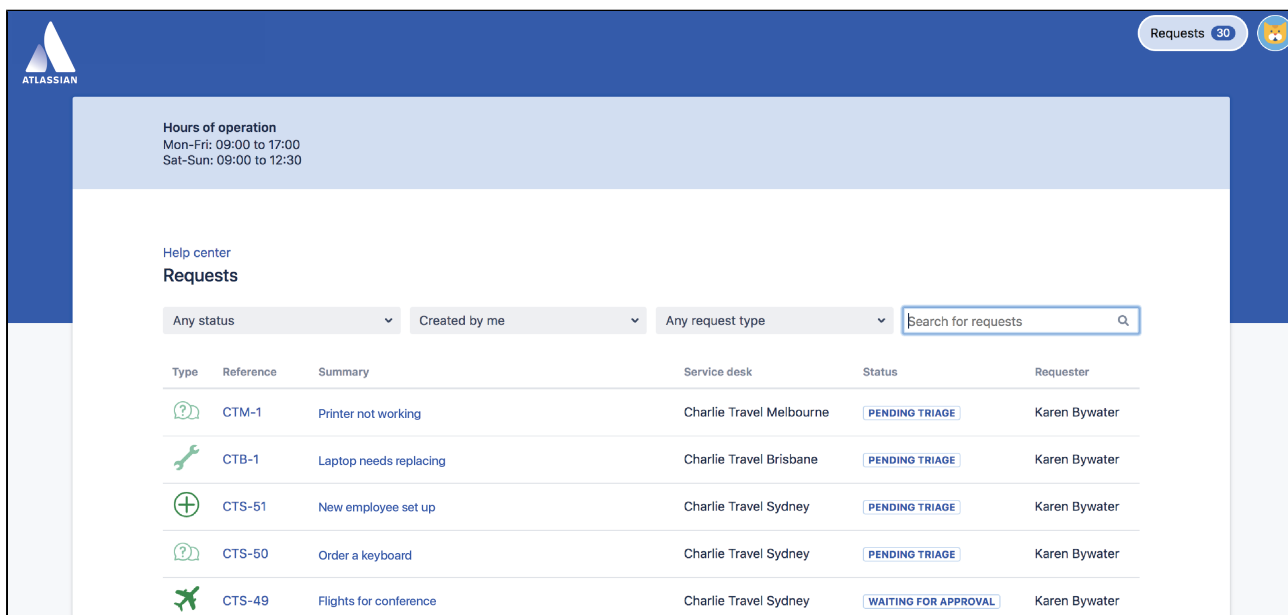
To enable comment reactions:

1. Go to **Project settings > Portal settings**.
2. Go to the **Comment reactions** section.
3. Select **Allow comment reactions on the customer portal for this project**.






View all customer portals in your help center

The help center shows all of the customer portals in your Jira Service Management instance. From here, customers can raise requests in any of the portals they have access to. They can also view all the requests they've raised, via the **Request** button in the top-right corner of the screen.

Check out [View requests in the help center](#) to learn how this works.



The screenshot shows the Jira Service Management help center interface. At the top, there's a blue header with the Atlassian logo and a 'Requests 30' notification. Below the header, there's a section for 'Hours of operation' with the following details: Mon-Fri: 09:00 to 17:00, Sat-Sun: 09:00 to 12:30. The main content area is titled 'Help center Requests' and features a search bar and several filter dropdowns: 'Any status', 'Created by me', and 'Any request type'. Below these filters is a table of requests with columns for Type, Reference, Summary, Service desk, Status, and Requester.

Type	Reference	Summary	Service desk	Status	Requester
	CTM-1	Printer not working	Charlie Travel Melbourne	PENDING TRIAGE	Karen Bywater
	CTB-1	Laptop needs replacing	Charlie Travel Brisbane	PENDING TRIAGE	Karen Bywater
	CTS-51	New employee set up	Charlie Travel Sydney	PENDING TRIAGE	Karen Bywater
	CTS-50	Order a keyboard	Charlie Travel Sydney	PENDING TRIAGE	Karen Bywater
	CTS-49	Flights for conference	Charlie Travel Sydney	WAITING FOR APPROVAL	Karen Bywater

Share the help center


To share the help center with your customers, send them the following link:

```
http://<computer_name_or_IP_address>:<http_port_number>/jira/servicedesk/customer/portals
```

If you add a customer to a service project, they'll have an account in the help center.

Receiving requests by email


If your customers prefer to raise requests from the comfort of their email inbox, you can set up an email address to receive requests in your service project. Every channel you create can be used to receive different request types. Emailed requests are added to your queues so your team can focus on customers without worrying about missing requests or handling multiple inboxes.

 You need a Data Center license to create multiple email channels for a service project.

Before you begin:

- Make sure you're a Jira administrator and read [Managing the email channel](#) to learn more about global mail settings.
- Enable [public signup](#), or [add customers](#) to your service project to ensure that you receive new requests.
- Set up a suitable [request type](#) with **Summary** and **Description** as required, visible fields. Any other fields must be optional.
- Learn which emails from your mail client will be [processed](#).
- If you want to set a Google or Microsoft email address for your project, you need to first create an OAuth 2.0 integration. This is necessary as Google and Microsoft disabled basic authentication (user and password). For more info, see [Configure an outgoing link](#).
- If you want to set a Microsoft GCC email address for your project, you need to select the Other email service provider type.

If you encounter any issues during the email setup process, check out [some common errors and resolutions](#).

 Customer requests and comments are processed differently than Jira mail. Issues created via Jira email handlers don't show up as service project customer requests. For this reason, we don't recommend using a Jira mail handler for service projects. Read [Creating issues and comments from email](#) for more information.

On this page:

- [How your customers send requests by email](#)
- [Add an email channel](#)
- [Verify your linked email channel](#)
- [Set up permissions so customers can send email requests](#)
- [How different types of emails are processed](#)

How your customers send requests by email

How receiving requests by email works:

1. A customer emails a request to an email address assigned to your service project. The request becomes an issue in your service project and is added to a queue based on the request type.
2. An agent comments on the issue.
3. The customer receives an email notification that contains the agent's comment.
4. The customer replies to the email notification and the reply is added as a comment on the issue in the service project.

Add an email channel


In Jira Service Management Server, you can add one linked email account to each service desk project.

Before you begin:

With Google and Microsoft deprecating basic authentication, you need to configure an OAuth 2.0 integration if you're using one of these email providers. The outgoing OAuth 2.0 link can be configured once by a Jira system admin, and then selected as an authentication method in your projects. For more info, see [Configure an outgoing link](#).

To add a new email:

1. From your service project, select **Project settings > Email requests**.
2. Select **Add an email address**.
3. If you're using Google or Microsoft as email providers, set the authentication method to the OAuth 2.0 integration configured in your Jira instance. You should see available OAuth 2.0 integrations in the dropdown list.

 When adding an email channel, you need to be able to authenticate to the email service provider used in your OAuth 2.0 integration.

4. Select a suitable request type for this channel.
5. Select **Save**. You'll be redirected to your email account.
6. Once authorized, you will see a success message about the channel being enabled. Your customers should now be able to create their requests via email.

Good to know

- If your Gmail or Yahoo! account uses two-step verification, you'll need to set up an [application-specific password](#).
- If you use more than one email address to interact with your customers, you might be able to set up forwarding rules or aliases to receive requests in the email linked to your service desk project. You will need to configure any forwarding rules or aliases in your email clients.

Verify your linked email channel

Once you've created a new email channel for a suitable request type, Jira Service Management sends a test email and creates a corresponding test request.

To verify your email:

1. From your service project, go to **Queues**.
2. Verify that your linked email channel works by finding the test request.

New messages sent to your linked email channel appear as service requests in your project.

Set up permissions so customers can send email requests

Before you share an email address with your customers, check your customer permissions.

- If [anyone can email your service](#) project, then your email address is ready to be shared. People who email your service project automatically become customers.
- If customers have to be manually added to your project, make sure your team [creates new customer accounts](#) for them. Otherwise, the customers' requests won't be processed.

Further reading

Check out [Managing the email channel](#) to learn how to:

- Control what happens to email addresses that are added to the **To** or **CC** field of an email associated with a service request
- Select whether to allow emails from addresses that aren't registered as customers of your service project to be added as comments to the associated request

How different types of emails are processed

Jira Service Management processes different emails differently. See the option that applies to you:

Emails using POP

Jira Service Management looks for messages in your inbox that:

- have the *Deleted* flag set to false
- were received after your email account and service project were linked

To link your email account using POP with a service project, make sure that your email inbox is empty by moving the existing messages to another folder, archiving, or deleting them. Starting with an empty inbox ensures that you do not lose emails unintentionally, as POP emails are deleted after they are processed by Jira Service Management.

Emails using IMAP

Jira Service Management looks for messages in your inbox that:

- have the *Deleted* and *Seen* flags set to false
- were received after your email account and service desk project were linked

If you use IMAP, emails are marked as read (not deleted) after they are processed by Jira Service Management. If you want existing messages to be pulled in by Jira Service Management, you can move them back to your inbox and mark them as unread after the connection has been established.

Managing the email channel

After you [set up your email channel](#), you can change how emails are parsed, if customers are created directly from emails, or view logs and check the email connection for any email channel on the site.

On this page:

- [Manage the email channel for multiple service desk projects](#)
- [Manage global mail settings](#)
- [How to configure a mail server and mail handler](#)

Manage the email channel for multiple service desk projects

You must be a Jira administrator to manage the email channel.

To manage the email channel:

1. Choose **Jira administration > Applications**.
2. Under **Jira Service Management**, select **Email requests**.
3. Choose one of the following:
 - **Strip quotes**: choose whether emailed comments contain the entire email thread, or just the most recent reply.
 - **HTML email parsing**: choose whether HTML emails display in wiki markup or plain text.
 - **Public email comments**: choose whether replies are added as comments to existing requests.
 - **Customer account creation**: choose whether new customers are created from the address list of email messages.
 - **View status and logs**: view the status of each email channel on your service project's site, logs, or delete a channel. Information older than 6 months is deleted daily.

Manage global mail settings

There are two global mail settings **email puller** and **email processor** that are used by Jira Service Management only. They don't impact any other email settings you have set up for Jira.

To manage global mail settings:

1. Choose **Jira administration > System**.
2. Under **Mail**, select **Global mail settings**.
3. Choose one of the following:
 - **Email puller**: connects to your mail servers every minute and pulls the email data into the database. Emails with attachments larger than 25MB are not pulled.
 - **Email processor**: filters the emails (for example, to remove auto-replies and spam) using information stored in the database.

How to configure a mail server and mail handler

To learn how to configure Jira Service Management to receive and process emails, check out [Creating issues and comments from email](#).

You must be a Jira administrator to do this.

Troubleshooting issues with the email channel

There are common errors when setting up an email channel for your service project. If you are not receiving requests or if you are getting errors, check the email connection. You might also review how emails are processed to see why a certain type of email isn't making it to your service project.

Check you email connection

To troubleshoot email channel issues, the first thing to do is to check the connection between Jira Service Management and your email account. You will see error messages that show you why the email channel doesn't work for your service project.

To check your connection:

1. Choose **Jira administration > Applications**.
2. Under **Jira Service Management**, select **Email requests**.
3. Select **Test**.

Check how you process email requests

Global mail settings control how your site processes requests that your customers send to your service project's email addresses. We process emails in three stages:

- **Email puller**: fetches unread emails from the service project's associated inbox every minute, and copies them to the email database. Emails with attachments larger than 25MB are not fetched.
- **Email processor**: filters pulled emails to remove things like auto-replies and spam.
- **Database cleaner**: deletes old emails from the database that are older than 45 days. You can change the purge to a range of intervals from 14 days to 180 days.

You can temporarily adjust these settings to troubleshoot your email channel.

To change global mail settings:

1. Choose **Jira administration > System**.
2. Go to **Mail > Global mail settings**.

Common email channel errors

This section describes the common errors and provides information about how to resolve them.

Setting up the email channel

Message:

`This authentication method is deprecated, you should use OAuth 2.0 instead.`

Resolution:

You're trying to use an email account from Google or Microsoft with the basic authentication credentials user and password. Since basic authentication is being deprecated by Google and Microsoft, you need to switch to the OAuth 2.0 authentication method. OAuth 2.0 can be configured by your Jira system admin for the whole Jira instance and then used in your projects. Once it's configured, you can choose it as a method from the drop-down list instead of providing the credentials. For more info on how to configure OAuth 2.0, see [OAuth 2.0 integration](#).

Message:

The email address you entered is currently used by another project's email channel. Please choose another email address. Check out our troubleshooting docs for help resolving the issue.

Resolution:

Make sure each of your service projects has a unique email account. An email account can't connect with multiple service projects.

You can also receive this message if you're trying to use multiple email aliases that point to the same email account for multiple service project email channels.

Requests aren't created or notifications received**Description:**

Customers are emailing requests that aren't arriving in your queues. They also aren't receiving notifications.

Resolution:

1. Use the instructions above to check the email connection.
2. Verify that the [customer has permission](#) to email the service desk.
3. Verify that the default request type for the email channel has visible **Summary** and **Description** fields.
4. Verify that the **Email puller** is on in **Global mail settings**.

No suitable request type for the email channel**Message:****No suitable request type for the email channel**

You will select the default request type assigned to requests created from the email channel during the setup. However no existing request types are suitable for email requests. A suitable request type must have both the Summary and Description fields as visible fields, and all the other visible fields, if any, must be optional. If you want to enable the email channel, add a new request type that meets the criteria or modify an existing one on the Request types page.

Resolution:

1. In your service project, select **Project Settings > Request types**.
2. Add a new request type or choose an existing one.
3. Select **Edit fields**.
4. Make sure both the **Summary** and **Description** fields are added and marked as **Visible**. Add an Attachment field and mark it as not Required.
5. Select **Save**, then head back to **Project settings > Email requests**.

Archived messages are creating requests**Description:**

When Jira Service Management checks your email accounts for new messages, it polls the inbox folder. Gmail uses labels to classify messages into categories and only has the following folders:

- Inbox
- Sent Mail
- Bin (or Trash)

This means that the archived messages are treated like they're in the inbox folder. With POP3, Jira Service Management is not able to identify archived messages by labels and therefore still brings them in as requests.

Resolution:

Use IMAP instead.

Setting up a Gmail account**Message:**

Unfortunately Jira couldn't connect to the mail server. Here is what the mail server said: "[ALERT] Please log in via your web browser: <http://support.google.com/mail/accounts/bin/answer.py?answer=78754> (Failure)

Resolution:

Create an [application-specific password](#) for Jira Service Management in your Gmail account settings.

Setting up a Yahoo! account**Message:**

Unfortunately Jira couldn't connect to the mail server. Here is what the mail server said: "[AUTHENTICATIONFAILED] (#MBR1240) Please verify your account by going to <https://login.yahoo.com>"

Resolution:

Create an [application-specific password](#) for Jira Service Management in your Yahoo! account settings.

Microsoft Outlook locks account for security**Message:**

Unfortunately Jira Service Management couldn't connect to the mail server. Here is what the mail server said: "STAT command failed: Exceeded the login limit for a 15 minute period. Reduce the frequency of requests to the POP3 server.

Resolution:

Jira Service Management checks email accounts every minute. Microsoft Outlook might suspect inappropriate usage of this account and lock it for security reasons. To resolve this, use IMAP instead.

Jira doesn't trust the SSL certificate on the the mail server**Message:**

Unfortunately JIRA couldn't connect to the mail server. Here is what the mail server said: "unable to find valid certification path to requested target" Check out our troubleshooting docs for help resolving the issue.

Resolution:

Make sure that the SSL certificate chain from the mail server is imported into Jira's Java trust store. Follow the steps on <https://confluence.atlassian.com/kb/unable-to-connect-to-ssl-services-due-to-pkix-path-building-failed-779355358.html>.

Setting up service project users

When you set up your service project, you add users to the project so that your team can start receiving and resolving requests. Your service project has the following users:

- **Project administrators** set up the service project and users.
- **Agents** work on customer requests and add customers to the project.
- **Customers** send requests to your service project.
- **Organizations** are groups of customers that are shared across projects.

By default, you need different permissions to manage different types of users in the project:

- **Jira administrators** can manage users and licenses across multiple projects.
- **Project administrators** can add agents from other projects. They can also manage customers and organizations.
- **Agents** can manage customers and organizations.

On this page:

- [Add agents to a project](#)
- [View roles of team members](#)
- [Remove agents from a project](#)
- [Unlicense agents](#)
- [What is a customer?](#)
- [Add a customer](#)
- [Remove customers from a project](#)
- [What is an organization?](#)
- [Add organizations](#)
- [Add customers to an organization](#)
- [Remove customers from an organization](#)
- [Remove an organization from a project](#)
- [Delete an organization from a site](#)
- [Involve Jira Software or Jira Core users](#)
- [Example of collaboration between Jira Service Management and Jira Software](#)
- [Learn more about managing users](#)

Add agents to a project

By default, project administrators can add agents from other projects to the project. Jira administrators can add anyone to the project.

To add agents:

1. From your service project, click **Add team** in the sidebar.
2. The agents are emailed a link to the service project and are added to the **Service Desk Team** project role.

View roles of team members

You must be a project administrator to view the roles of team members.

To view the roles of each team member:

1. From your service project, select **Project settings > Users and roles**.
2. Select the **Roles** dropdown, or search for the team member directly.

When a Jira administrator adds a new agent to the project, the agent is also assigned a Jira Service Management license and added to the **service-desk-users** license group.

To view what license or group a user belongs to:

1. Go to **Jira administration > User management**.
2. Search for and select a user to view their profile.

If your agents need to collaborate with Jira Core or Jira Software users to resolve an issue, you can grant the Jira users limited access to your service project. Learn more at [Involve Jira Software and Jira Core users](#).

Remove agents from a project

You must be a project administrator to remove an agent from a project.

To remove an agent from a project:

1. From your service project, select **Project settings > Users and roles**.
2. Hover over the user or group you'd like to remove from the **Service Desk Team** project role, then click the trash can icon to the right.

Unlicense agents

Jira administrators can remove an agent's license.

To unlicense an agent:

1. Go to **Jira administration > User management**.
2. Select the user.
3. Under the **Actions** column, choose **Edit user groups**, then remove them from the **Jira-service-desk-users** group.

What is a customer?

Customers are people who can send requests to your service project via the portal or email. How people become customers depends on your project's customer permissions. To see your customers, go to **Customers** from your service project.

Customers are automatically added to the list if your project is open to users with Jira accounts, or allows customers to create their own accounts. Customers who do not have Jira accounts are added to the **Service Desk Customers** project role in **Project settings > Users and roles**. Customers are also granted restricted access to the Customer portal only (not Jira).

For more information about how people become customers, see [Managing access to your service project](#).

Add a customer

Both agents and project administrators can add customers to projects.

To add customers:

1. From your serviceproject, go to **Customers**.
2. Click **Add customers** to add customers to your projects.

Remove customers from a project

You must be a project administrator to remove customers from a project.

To remove a customer from a project:

1. From your service project, select **Project settings > Users and roles**.
2. Hover over the customer you would like to remove from the **Service Desk Customers** project role, then click the trash can icon to the right.

If the customers have Jira accounts or created their own accounts, then a Jira administrator needs to deactivate them.

To deactivate a user:

1. Go to **Jira administration > User management**.
2. Search for and select the user you want to deactivate.
3. Under the **Actions** column, select **Edit**.
4. Uncheck the box beside **Active** to deactivate the user.

What is an organization?

Organizations are groups of customers that can be used in multiple projects. When you add an organization to a project, its members can raise requests in the project and share them with the organization. They're also notified about the organization's requests and can view and search them on the **My Requests** page in the portal. Learn more about [how customers share requests with organizations](#).

By default, you need the **Service Desk Team** role for a project to manage organizations in it. However, a Jira admin can restrict organization management to Jira admins by turning off the Organization management setting:

1. Go to **Jira administration > Applications**.
2. Under **Jira Service Management**, select **Configuration**.
3. Look for the **Organization management** heading.
4. Organizations are managed from a project's **Customers** list.

Add organizations

Organizations are groups of customers that can be used in multiple projects.

You must be a project administrator to add organizations to the project.

To add an organization:

1. From your service project, go to **Customers**.
2. Select **Add organizations**.
3. Add a new organization, or add an existing organization.

The organizations that you add display on the **Customers** list.

Add customers to an organization

Rather than add customers individually, you can group customers into organizations which can be part of multiple projects. Customers can then raise requests in all service projects they have access to.

You must be a project administrator to add customers to an organization.

To add customers to an organization:

1. From your service project, go to **Customers**.
2. Select the organization you want to add customers to.
3. Click **Add customer**.

If the customers are new to the Jira site they are given restricted access to the Customer portal only (not Jira). Also, customers are not added to the **Service Desk Customers** project role, because the organization can be used in multiple projects. However, they can still raise requests in all projects that use the organization.

Remove customers from an organization

Customers removed from organizations lose access to projects that use the organization, unless they have access through another organization, have the **Service Desk Customers** project role, or the project allows anyone to raise a request without logging in.

You must be a project administrator to remove customers from an organization.

To remove customers from an organization:

1. From your serviceproject, go to **Customers**.
2. Select an organization in the **Customers** list.
3. Find the customer you want to remove, and then click **X** to the right.

Remove an organization from a project

When you remove an organization from a project, its members lose access to the project unless they have the **Service Desk Customers** project role, have access through another organization, or the project allows anyone to raise a request without logging in.

You must be a project administrator to remove an organization from a project.

To remove organizations from a project:

1. From your service project, go to **Customers**.
2. Select an organization in the **Customers** list.
3. Click **Remove from project**.

Delete an organization from a site

Deleting an organization does not delete the customers who are in it. The customers still exist on the Jira site and have access to any projects for which they have the **Service Desk Customers** role or access via another organization.

You must be a Jira admin to delete an organization from a Jira site.

To delete an organization from a site:

1. From your service project, go to **Customers**.
2. Select an organization in the **Customers** list.
3. Select **Remove from project > Delete organization**.

Involve Jira Software or Jira Core users

You can give users with Jira Software or Jira Core licenses permission to view and comment on service project issues without a Jira Service Management license, **only if Jira Core, Jira Software, and Jira Service Management are licensed on the same Jira instance.**

To involve Jira application users:

1. From your service project, select **Project settings > Users and roles.**
2. Add users to the **Service Desk Team** role.

These users can:

- View issues, comments, and attachments
- Add and delete their own attachments and internal comments
- Watch and vote for issues

These users can't:

- Leave comments for customers
- View queues, the customer list, or reports
- Transition service project issues
- Log work on a service project issue
- Be assigned to a service project issue

Learn more about [collaborating with Jira Software or Jira Core users.](#)

Example of collaboration between Jira Service Management and Jira Software

1. Martin, an IT service desk team agent, links an incident ticket in a serviceproject to an underlying network problem ticket in a Jira Software project.
2. Andrew, a Jira Software developer on the network operations team, assigns this network issue to himself and starts working on it.
3. After fixing the problem, Andrew opens the linked serviceincident ticket and leaves an internal comment asking Martin to try the network connection again.
4. After receiving the internal comment, Martin verifies the network connection and tells the customer that the problem is resolved.

Learn more about managing users

Check out the following documentation to learn more about managing users and permissions:

Documentation	Details
User management	Add and remove users, manage users with groups, and manage access to Jira applications.
Managing project roles	Add and remove project roles and manage project role membership.
Enabling public signup	Allow customers to create their own accounts by signing up on the customer portal or emailing the email channel
Configuring permissions	Learn how global permissions affect licensing, how project permissions are associated with a project role, and how to customize the default service desk project permission scheme.
Customizing Jira Service Management permissions	For advanced admins, find out more about service project users' required permissions.

Resolving Jira Service Management permission errors	Troubleshoot permission errors.
---	---------------------------------



Managing project role memberships


You can use project roles to easily associate users and groups with a particular project. For example, you may want to send notifications to a specific set of users associated with your project, and by adding them all to a project role, you can then use that project role to control who receives the notifications.

You can also use project roles to restrict how much access certain users or groups have. Unlike groups, which have the same membership throughout your application, project roles have specific members for each project.

This page contains instructions for managing membership of *existing project roles*. For information on creating and using project roles, see [Managing project roles](#).

Viewing and editing project role members

1. Log in as a project administrator and open your project.
2. Select **Project settings** () > **Users and roles**.
3. You'll see all users and groups associated with each project role.
4. To add users or groups to a project role, select **Add users to a role** in the top right corner. Enter the users or groups and select the project role you wish to add them to.
5. To remove a user or group from a project role, hover over the user or group row, and select **Delete** ()

 Since group membership can only be edited by users with the **Jira Administrator** global permission, project administrators may therefore prefer to assign users, rather than groups, to their project roles.

Setting up queues for your team

Make sure that your team is working on the right requests at the right time with easily configurable service project queues. A queue is a filtered set of issues that are displayed to your team. Use your project's default queues or create custom queues to save time triaging requests, and to give your agents more visibility of the number and type of incoming customer requests they need to work on.

You need to be an administrator for your project to set up queues.

On this page:

- [Creating new queues](#)
- [Managing queues](#)
- [How your team uses queues](#)

You can set up queues on the aptly named **Queues** (📄) tab in your service project:

Time to resolution	T	Key	Status	Summary	Created	P	Reporter
-3d	🔴	CTF-1	WAITING FOR SUPPO...	What am I looking at?	26/Apr/18	🔴	Karen Bywater
1h 19m	🟢	CTF-2	WAITING FOR APPRO...	3 day trip to Brisbane	02/May/18	🟢	Alana Grant

Creating new queues

When creating a new queue, you can select the queue name, the issues that will be filtered into this queue, and the columns that appear in the queue to make life easier for your service team. Here's how you create a new queue:

1. From your service project sidebar, select **Queues > New queue**.
2. Name your queue using language your team will understand (for example, "Due this week").

3. Select which issues will show up in this queue using the dropdown options in the Basic search view:

You can also select the advanced search view to enter a [Jira Query Language \(JQL\)](#) statement.

4. Add or remove columns to control what issue information, such as the issue key and issue creation date, is displayed in your queue.
5. Select **Create**. If you have existing issues in your project that fit the criteria selected in "Issues to show", these issues will now appear in your new queue.

Managing queues

You can reorder or delete queues at any time by hovering over the Queues sidebar and selecting **Manage**. In the Manage queues dialog that appears, you can see the number of issues in each queue, and drag and drop queues to reorder them.

You can edit existing queues by selecting the queue you wish to configure and selecting **Edit queue** in the top right corner. You can edit the queue name, the issues shown, the columns, and column order. Note that you will see a live preview of the updated issues that appear in this queue as you configure it.

i If you're using filters in your queues, make sure these filters include all the priorities defined in the associated priority scheme. See [Associating priorities with projects](#) for more details.

How your team uses queues

Queues give your agents a single view of the work that needs to be done across their team. Agents can view all of the queues in your service project; however, they cannot create or edit queues.

Automating your service project

You're viewing Legacy automation

We've renamed **Automation** to **Legacy automation** after introducing new automation engine to Jira Data Center. The page you're viewing describes original automation rules that have been part of Jira Service Management since the beginning. You can still use them and your existing automation rules will keep working. We recommend, however, that you switch to the more powerful Automation.


To check out the new Automation, see [Automate your project](#).

On this page:

- [Set up a preset automation rule](#)
- [Edit preset automation rules](#)
- [Create a custom automation rule](#)
- [Customize automation webhooks](#)
- [Rule run options](#)
- [Disable an automation rule](#)

Create automation rules to perform actions in your service project based on specific triggers and conditions. For example, you can set an automation rule that alerts an agent when a high priority issue is created. Or, service project can reopen an issue if your customer comments on it after its been resolved.

Set up a preset automation rule

 If your automation rules involve priorities and you're using filters in your service project, make sure these filters include all the priorities defined in the associated priority scheme. See [Associating priorities with projects](#) for more details.

To set up an automation rule:

1. In your service project, select **Project settings > Legacy automation**.
2. Select **Add rule**.
3. Select a preset rule from the list, then select **Next**. The rule configuration screen appears.
4. Edit the rule name and description as needed. The rule name appears on the main automation settings page, so select a name that explains what it does.
5. Edit and update any fields that appear in the configuration dialog.
6. Select **Save**.

Your service project comes with preset rules that you can use to set up automation. Here are the preset rules that come out-of-the-box:

Preset rules
<p>Transition on comment</p> <p>When a comment is added to an issue, this rule automatically transitions the issue so it's clear who it's waiting on.</p>
<p>Reopen on customer comment</p> <p>When a comment is added to a resolved issue, this rule will reopen the issue.</p>
<p>Be aware of urgent issues</p> <p>This rule alerts you (via an @mention) to urgent issues when they are raised, so you can address them immediately.</p>
<p>Keep on top of SLAs</p> <p>This rule alerts you(via an @mention) when a serious issue is about to breach one of your SLAs.</p>
<p>Set customer expectations</p> <p>This rule lets your customers know when to expect a response based on the priority of their ticket, by adding a prepopulated comment to the issue.</p>
<p>Update when a linked issue changes</p> <p>This rule will add a comment to related issues, when the status of an issue changes.</p>
<p>Triage requests sent by email</p> <p>This rule updates issues received by email with the correct request type, based on keywords present in the request summary or description.</p>

Edit preset automation rules

To edit a preset rule:

1. In your service project, select **Project settings > Legacy automation**.
2. Select **Edit** next to the rule in your automation list to see how it's configured.
3. Select the **WHEN, IF, or THEN** fields to change your rule's trigger, conditions, or resulting actions. Use **Tips for customizing this rule** for suggestions on what to enter in these fields.
4. Select **Save** to publish your changes.

Create a custom automation rule

To create a custom automation rule:

1. In your service project, select **Project settings > Legacy automation**.
2. Select **Add rule**.
3. Select **Custom rule** from the list, then select **Next**. The rule configuration screen appears.
4. Give your custom rule a name and a description.
5. Configure your rule by defining the **WHEN, IF, and THEN** fields.
6. Select **Save**.

Here are the rule sets that are allowed in the automation engine:

WHEN	IF (optional)	THEN
Comment added	<p>Issue matches a certain filter</p> <p>Comment visibility is internal or external</p> <p>User type is a customer or agent</p> <p>Comment contains a key phrase</p>	<p>Add comment, either internal or external</p> <p>Alert user to prompt a specific user or users via an @mention</p> <p>Archive issue to remove it from search results, queues, the customer portal, and all other places it previously appeared. You can access archived issues through direct links, and restore if needed. Learn more</p> <p>Auto-approve/decline to approve/decline a request based on your IF field</p>
Comment edited	<p>Issue matches a certain filter</p> <p>Comment visibility is internal or external</p> <p>User type is a customer or agent</p> <p>Comment contains a key phrase</p>	<p>Edit request type to change the request type (Because request types are mapped to specific issue types, automation isn't able to change issue types. Be sure your request types are the same issue type before applying this rule)</p> <p>Edit issue to select and change a field in your issue, like assignee or priority, as well as Jira supported custom fields (see Jira administration > Issues > Custom fields).</p> <p>Send email to create an email notification.</p> <p>Transition issue to change its position in the workflow</p> <p>Webhook to send a POST request.</p>
Issue is created	<p>Issue matches a certain filter</p> <p>User type is a customer or agent</p>	
Issue resolution is changed	<p>Issue matches a certain filter</p> <p>User type is a customer or agent</p> <p>Resolution change is either set or cleared</p>	
Status changed	<p>Issue matches a certain filter</p> <p>User type is a customer or agent</p> <p>Status change visible to customer</p>	

<p>A linked issue is transitioned</p>	<p>Link type matches a certain type of link (for example, is related to or blocks)</p> <p>Issue matches a certain filter</p> <p>Linked issue matches a certain filter</p> <p>User type is a customer or agent</p>
<p>Request participant added</p>	<p>Issue matches a certain filter</p> <p>User type is a customer or agent</p>
<p>Organization added to issue</p>	<p>Issue matches a certain filter</p> <p>User type is a customer or agent</p>
<p>Approval required</p>	<p>Issue matches a certain filter</p> <p>User type is a customer or agent</p>
<p>SLA time remaining</p> <p>Select the SLA and goal status that triggers the event</p>	<p>Issue matches a certain filter</p>

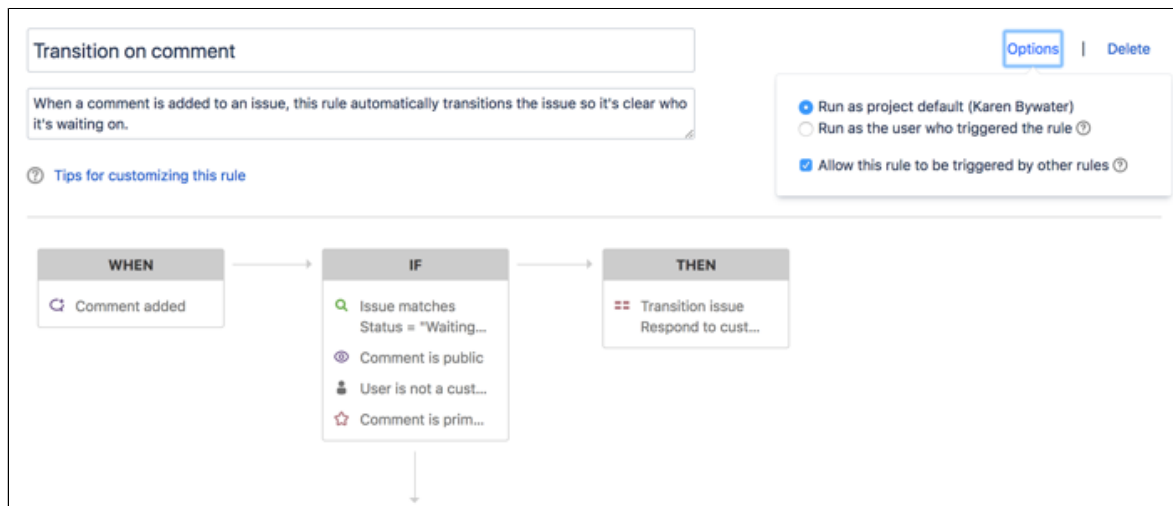
Customize automation webhooks

Jira Service Management webhooks let you send information to third party web services without the need for a third party app. For example, you can post a message in a Slack channel when a major incident occurs.

Read [Send alerts with Jira Service Management webhooks](#) to learn more.

Rule run options

Select **Options** in the rules configuration page to change the behavior of how a rule runs, as shown in the image below:



Run rule as: By default, rules run as the person who created the project. Alternatively, you can run rules as the person who triggers the rule. For example, if a rule responds to a customer, then you might want the comment to be from the agent who is working on the request, not from the person who created the project. In this case, you would want to set the **Run rule as** to **User who triggered the rule**. Make sure the person you choose has permission to complete all the actions taken by the rule.

Triggered by other rules: By default, rules can trigger other rules. In some cases, you might need to disable this to prevent two rules from triggering each other infinitely.

To change the default **Run as** user, go to **Legacy automation > Change default event user**. Make sure the person you choose has permission to perform all the actions that you might try to automate in all your rules.

Disable an automation rule

To disable an automation rule:

1. Go to **Project settings > Legacy automation**.
2. Select **Edit** next to the rule in your automation list. The rule configuration screen appears.
3. Unselect the **Enable rule** checkbox (at the bottom) and select **Save**.

Disabled rules appear in your automation list with a **DISABLED** badge.

Send alerts with Jira Service Management webhooks

Webhooks are ways for one application to send automated information to another. You might use a webhook to communicate something important after a particular event has occurred. For example, you can send an SMS to a team lead when a high priority issue is created.

A webhook is made up of two things:

- **payload** the message itself
- **URL** the destination of the message. This URL needs to be [allowlisted in Jira](#).

By default, we will send payloads in JSON format using an HTTP POST request.

On this page:

- [Add a webhook to a Jira Service Management automation rule](#)
- [Payload body options](#)

Add a webhook to a Jira Service Management automation rule

A Jira Service Management webhook is a **THEN** action in an automation rule. When you configure a rule you can specify **WHENs** and **IFs**, then use the webhook to define your action.

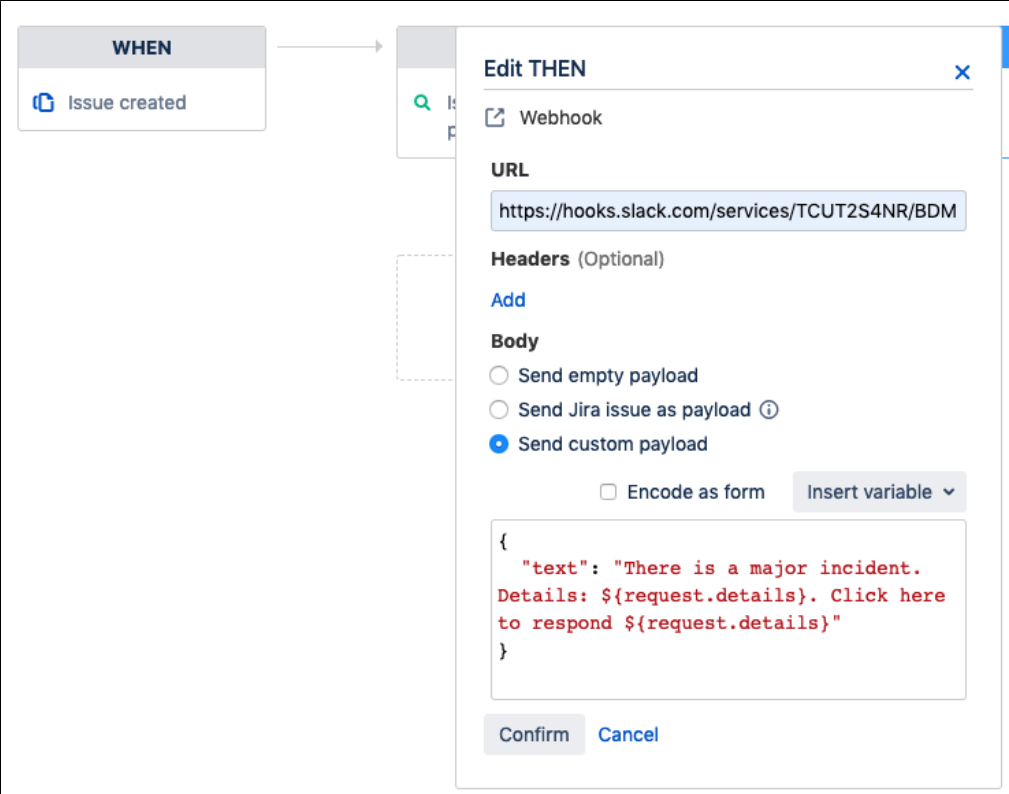
Before you begin:

- The destination URL needs to be [allowlisted in Jira](#).

To add a webhook to a Jira Service Management automation rule:

1. Go to **Project settings > Automation**.
2. Create a Custom rule or edit an existing rule.
3. Configure the **WHEN** and **IF** settings as you like.
4. Add a **THEN** action and choose **Webhook** from the dropdown.
5. Configure the webhook settings, name your rule, then hit **Save**.

See the example below of an automation rule that posts a message to Slack when a major incident occurs:



The screenshot shows the configuration for a Jira Service Management automation rule. On the left, under the 'WHEN' section, the trigger is set to 'Issue created'. An arrow points to the 'THEN' section, which is currently empty. A modal window titled 'Edit THEN' is open, showing the configuration for a 'Webhook' action. The 'URL' field is populated with 'https://hooks.slack.com/services/TCUT2S4NR/BDM'. Under 'Headers (Optional)', there is an 'Add' button. Under 'Body', the 'Send custom payload' option is selected. There is an 'Encode as form' checkbox and an 'Insert variable' dropdown. The custom payload is a JSON object:

```
{  "text": "There is a major incident. Details: ${request.details}. Click here to respond ${request.details}"}
```

 At the bottom of the modal are 'Confirm' and 'Cancel' buttons.

✔ Read the [developer tutorial](#) for detailed information on how to configure Jira Service Management webhooks.

Payload body options

You can choose one of the following three options for the payload's request body:

- **Send empty payload** - you might choose this option if you're communicating with an app that requires no information.
- **Send Jira issue as payload** - the Jira issue, user, and any comments will be sent.
- **Send custom payload** - you can choose the variables to include and customize your payload you'll need to specify your payload in JSON format. Check the **Encode as form** box if you want the payload to be URL-encoded.

Managing service project notifications

Customers, agents, and admins get email notifications about the status of, and activity on, requests. You can customize the recipients, content, and style of these notifications. This page is about customizing notifications that go to your customers.

To customize the notifications that go to agents and admins, make changes to the [Jira notification scheme](#).

On this page:

- [How different roles receive notifications](#)
- [Account verification emails](#)
- [Opt in and out of customer notifications](#)
- [Disable the customer scheme for customer notifications](#)
- [Edit customer notifications](#)
- [Automatically send emails as customer notifications](#)
- [Translate customer notifications into different languages](#)
- [Customize customer notifications](#)
- [Select HTML or plain text](#)

How different roles receive notifications

A service project sends notifications based on two schemes:

- One for your customer, and
- One for your licensed users, who are most likely agents and admins

Customers

When a customer submits a request, they receive the same default email notifications as the issue reporter or participant. When customers share requests in an organization, customers in that organization receive notifications. They can [opt in and opt out of notifications](#) for other activity on the request.

When an approver needs to approve the request, they'll receive a notification. They can opt in to notifications for other activity on the request.

Other customers receive notifications on public activities for requests they're involved with. They can [opt in and opt out of notifications](#) in the customer portal or email. Reporters who decline are only notified when a team member resolves a request.

When a customer first contacts a service project that has incoming email and public sign-up enabled, an account is set up for them in the background. They'll receive an email that asks them to reset their password and verify their account details. Project administrators can [disable these account verification emails](#).

Agents and admins

When agents and admins work on a request, they receive notifications as part of the [project's notification scheme](#).

Agents don't receive notifications on their own changes when they act as a customer on issues with a set request type. Jira Service Management treats agents acting as a reporter, participant or approver on these issues as a customer. This occurs regardless of the notification scheme.

Agents who are part of an organization will only receive a customer notification when a request is shared with that organization. If the agent is the assignee of the issue, they won't receive any notifications from the project's Jira notification scheme.

Account verification emails

When a customer raises a request by email for the first time, the following events take place:

- An account is created for the customer in the backend.
- A notification is sent to the customer asking them to verify their email address and reset their password, so they can access the customer portal.

This is confusing for customers of organizations running email-based service projects because, most likely, only the agents and admins are using the customer portal. The good news is that you can disable account verification emails to avoid customer confusion you need to be a project administrator to do this.

To disable account verification emails:

1. Go to **Project settings > Customer notifications**.
2. Navigate to the heading **Account verification email**.
3. Enable the **Do not send account verification emails** toggle (displaying as green with a tick).

Opt in and out of customer notifications

Customers receive notifications when other customers share requests with them. They can opt in and opt out of these notifications in the customer portal, or in email.

To opt in and opt out of notifications in the customer portal:

1. From your customer portal, select **Requests > All requests**.
2. Select the request whose notifications you want to opt in or opt out of.
3. Select either **Get notifications** or **Don't notify me**, besides the eye icon.

To opt out of notifications in email, select the link **Turn off this request's notifications** at the bottom of the email.

Disable the customer scheme for customer notifications

If you have an internal service project where all your help seekers have Jira licenses, you might prefer to disable the customer scheme. In this case, you can work from the licensed user scheme.

To disable the customer scheme:

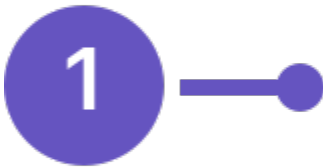
1. Go to **Jira administration > Applications > Jira Service Management > Configuration**:
2. Under the heading **Notifications**, select **Yes**. This tells your project to stop blocking Jira notifications for customers.
3. Disable the service desk notifications in **Project settings > Customer notifications**.
4. For each rule, select **Edit** and deselect the **Enable** checkbox at the bottom.
5. Go to your [Jira notification scheme](#) and customize the notifications for your customers, agents, and admins.

Edit customer notifications

To edit the recipients or message content of your service project notifications:

1. From your service project, select **Project settings > Customer notifications**.
2. Select the notification you want to edit, then select **Edit** under the **Action** column.

3. [Select recipients](#) by selecting and editing the **To** field. You can delete recipients by selecting the **X** next to an added recipient.
4. Under **Content**, edit the message. You can [add variables](#) using the **Insert variable** dropdown to pull blocks of information from issues.
5. To disable a notification, untick the **Enable** checkbox.
6. Select **Save** to confirm your changes.



Customer notifications

Request created

To

Customers involved x

Exclude person who caused the action

Content

English (United States) [Default] v

B *I* U

Just confirming that we got your request. We're on it.



[Formatting help](#)

[Revert to default content](#)

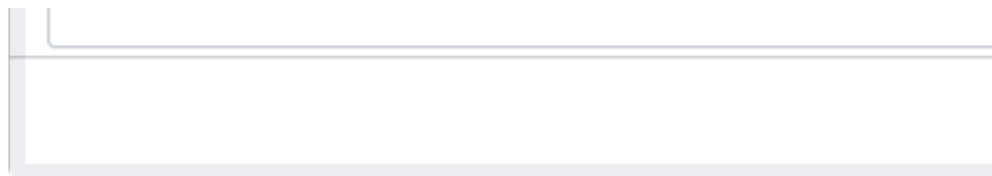
Preview

Just confirming that we got your request. We're on it.

[View request](#) · [Turn off this request's notifications](#)

This is shared with Alpha, Brett Kennedy, Rey Jakku, and Finn 218

Help center, powered by [Jira Service Desk](#), sent you this message



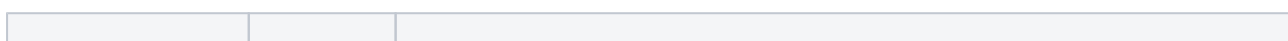
Select recipients

When choosing recipients, add at least one of the following to the To field:


Recipient options	Details
Reporter (customer)	The reporter of the request. This notification is sent even if the reporter has opted out of receiving notifications on the customer portal.
Customers involved	All customers involved in the request, including: <ul style="list-style-type: none"> • the reporter • the request participants • the people in the organization that the request has been shared with • the approver, in case the request requires an approval Notes: <ul style="list-style-type: none"> • This notification isn't sent to customers if they have turned off notifications for an individual request in the portal or a request's email thread. • For the users which fall into the category of people in an organization that the request has been shared with as well as approvers, this notification is turned off by default. This means that to receive notifications meant for the recipient of Customers involved and for a specific request, they need to explicitly turn on notifications for that request by selecting Get notifications.
Added participants	The people who have been added as request participants.
Added organizations	The people in an organization that the request has been shared with.
Approvers	The people who need to approve or decline a request.
Exclude person who caused the action	Prevents the person who triggered the rule from receiving a notification. For example, if a customer adds a comment to a request, you can select to exclude them so they don't receive a notification about their own comment.

Include issue variables

You can use variables to pull blocks of information from issues and insert them into your message. Select the **Insert variable** drop down menu to add valid variables for the notification you're customizing.



Name	Format	Description
Recipient name	<code>\${recipient.name}</code>	The full name of the person receiving the email.
Name of person who caused the action	<code>\${event.user.name}</code>	The full name of the person who triggered the notification, for example by adding a comment.
Issue summary	<code>\${issue.summary}</code>	The summary of the issue, or blank if there's none.
Issue description	<code>\${issue.description}</code>	The description of the issue, or blank if there's none.
Issue key	<code>\${issue.key}</code>	The issue key (for example, IT-123).
Issue reporter	<code>\${issue.reporter.name}</code>	The full name of the user that reported the issue.
Issue resolution	<code>\${issue.resolution}</code>	The resolution of the issue, for example "done".
Request URL	<code>\${request.url}</code>	The URL of the request in the customer portal.
Comment	<code>\${comment}</code>	The comment added to the issue. This is only available using the "Comment added" or "Comment edited" WHEN trigger.
Request status	<code>\${request.status}</code>	The customer-visible status of the request, as shown on the customer portal.
Portal name	<code>\${portal.name}</code>	The portal name, which can be edited on the Portal settings page.
Request details	<code>\${request.details}</code>	The full details of a request. This includes the creation date, request type, summary, and the same fields configured as visible in the request type setting. This does not include fields created by third-party add-ons.
Approval buttons	<code>\${approval.buttons}</code>	The Approve and Decline buttons to action requests from within the email.

 Fast-track the time it takes approvers to action pending requests by inserting **Request details** and **Approval buttons** to your approval notifications template, if they aren't there already.

Automatically send emails as customer notifications

To create custom email triggers:

1. From your service project, select **Project settings > Automation**.
2. Select **Add rule > Custom rule > Next**.
3. Edit the name and description as needed.
4. Edit these fields to configure your rule:
 - **WHEN**: select what triggers your rule

- **IF:** select the conditions of your rule
 - **THEN:** send email
5. Select **Save** to create your custom rule.

Translate customer notifications into different languages

Jira Service Management lets you provide translations and regional messages to your customers, and supports a large number of languages.

Read [Translate service project notifications](#) to learn how to do this.

Customize customer notifications

Your service project adds a default subject line that you can customize to better fit your brand.

To customize notifications:

1. From your service project, select **Project settings > Customer notifications**.
2. Under **Templates**, select **Edit templates**.
3. Under **Subject**, edit your subject line. You need to keep the issue key somewhere in the subject line to send a request's notifications into a single email conversation.
4. Under **Styles**, adjust the HTML, CSS, and plain text styles as you like. You can include more template variables by selecting the Insert variable drop down.

Include more information using template variables

You can use this set of variables to pull blocks of information and add it to your email template.

Name	Format	Description
Message content	<code>\${message.content}</code>	The content of your notifications. We require this variable. This variable includes batched messages.
Request location	<code>\${request.url}</code>	The URL of the request.
Disable notifications location	<code>\${request.disable.notifications.url}</code>	The URL to turn off notifications for the request. As a best practice, we recommend always including this link.
List of request viewers	<code>\${request.sharedwith}</code>	The participants, approvers and customer organizations who can also view the request.
Help center name	<code>\${helpcenter.name}</code>	The name of your site's help center, an entry point to all customer portals on your site.
Jira Service Management homepage	<code>\${atlassian.url}</code>	The URL of Jira Service Management's homepage.

Style more CSS classes

You can style more classes than appear in the default template.

Name	Class	Description
Reply marker	<code>.jsd-reply-marker</code>	The style of the reply marker. This series of dashes and hyphens is used to process comments when a customer replies to a notification.
Batched item container	<code>.jsd-activity-item-content</code>	The style of a division used to wrap individual messages added to a batched notification.
Batched message separator	<code>.jsd-activity-item-separator</code>	The style of a separator inserted when batches of messages are sent as a single notification.

Select HTML or plain text

If the default type is set to HTML, we send dual-encoded notifications. This means your customers can then select the HTML or plain text view in their mail client. If your customers use a plain text mail client, change your default setting and apply it to new and existing customers.

You must be a Jira administrator to set the default email type.

1. Select **Project administration > System**.
2. Under **User Interface**, select **Default User Preferences**.
3. Select **Edit default values**.
4. Change the **Default outgoing email format** to HTML or text and select **Update**. At this point, the email format you have selected will only be applied to new service project customers.
5. If you also want to override the email format chosen by existing service project customers and agents, select **Apply** under **Actions**.
6. Select **Update** to finish applying the email preference to all user accounts.

Translate service project notifications

When you create a project, its help center and customer portal display in the default language from your Jira instance.

To support your customers who speak different languages, you can translate the notifications they receive like the confirmation email that lands in their inbox after they submit a request.



Check out [Managing service project notifications](#) to learn:

- how different roles receive notifications
- how to customize notifications

On this page:

- [Add or remove a language](#)
- [Review and translate customer notifications](#)

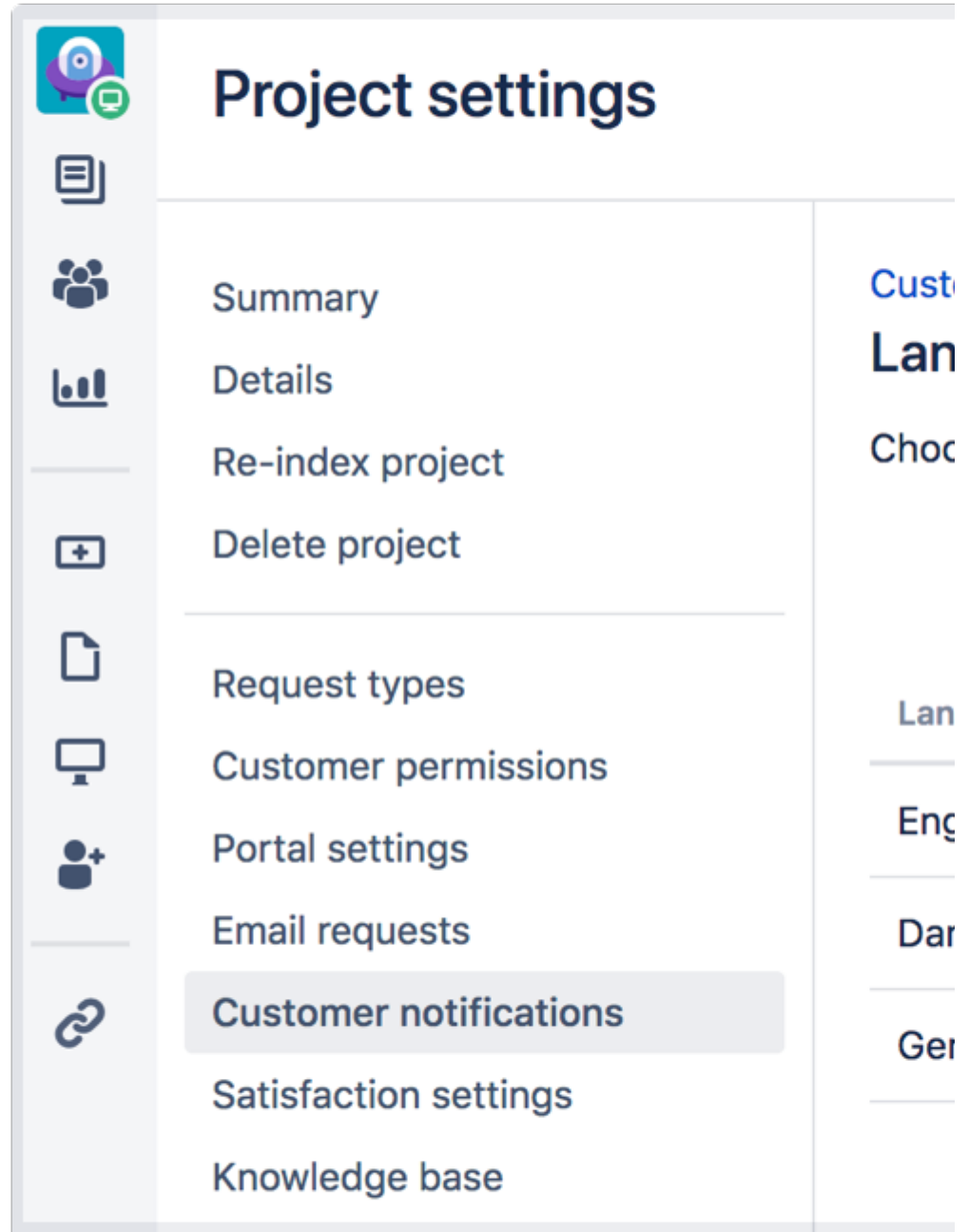
Add or remove a language

When you add a language, your service project introduces translated notification messages and templates from our default style.

To add or remove a language to your service project:

1. Go to **Project settings > Customer notifications**.
2. Under **Language support**, select **Manage languages**.
3. Select **Add languages**, the languages your service project supports appear in the table.
4. Choose the language(s) you'd like to translate.
5. Under the **Action** column, you can **Enable** or **Disable** languages. We disable newly added languages by default, so you can quality check your translation before enabling it.

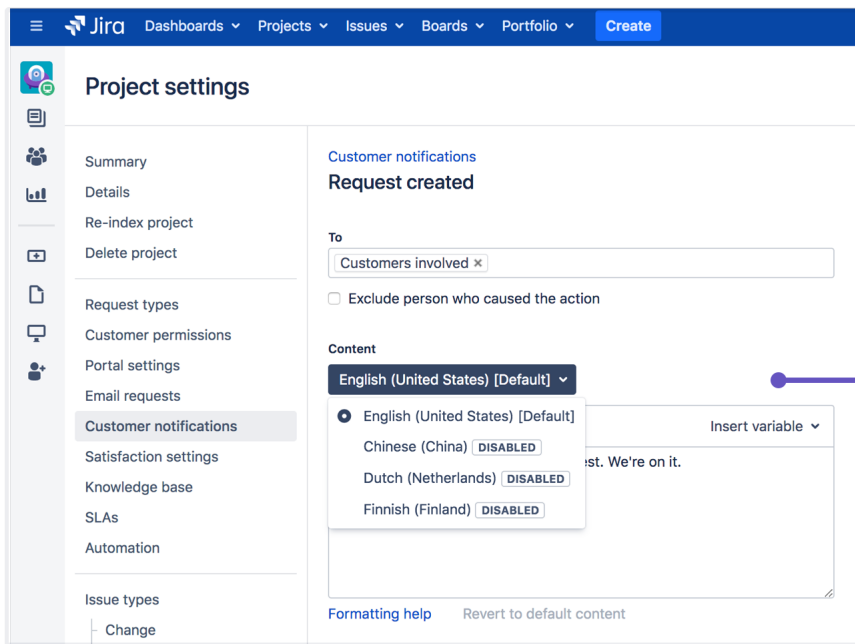
If a language isn't supported in your service project, your customers receive notifications in the project's default language. You'll see a project default badge beside the default language, and you can change this via the **Set project default language** link.



Review and translate customer notifications

To translate a customer notification:

1. Go to **Project settings > Customer notifications**.
2. Under **Default rules**, choose the notification to translate, then select **Edit**.
3. On the notification screen, under **Content**, select a language from the drop down.
4. Insert any additional variables or custom content. You'll need to add translations for your custom content, otherwise, customers will see that content in the project default language.
5. Preview the content, then hit **Save**.



Setting up request types

Jira Service Management provides a set of default request types that are configured for basic IT help desk scenarios. You can configure the default request types or add new ones to suit the needs of your customers and team. Request types can be organized into groups to help customers find the request they need on the customer portal.

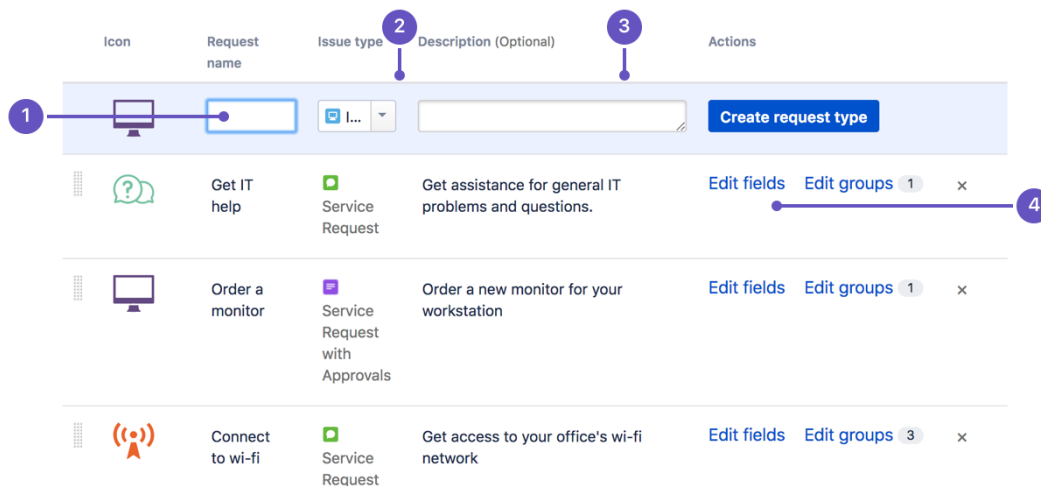
i You need to be an administrator to set up request types and workflows in your project.

On this page:

- [Set up request types](#)
- [Organize request types into groups](#)
- [Customize the fields on a request type](#)
- [Customize the workflow statuses for a request type](#)
- [Hidden fields and unsupported fields](#)
- [Set up URLs with auto-populated request fields](#)

Set up request types

Each request type in a service project is based on an issue type. Open **Project settings > Request types** to manage your project's request types:

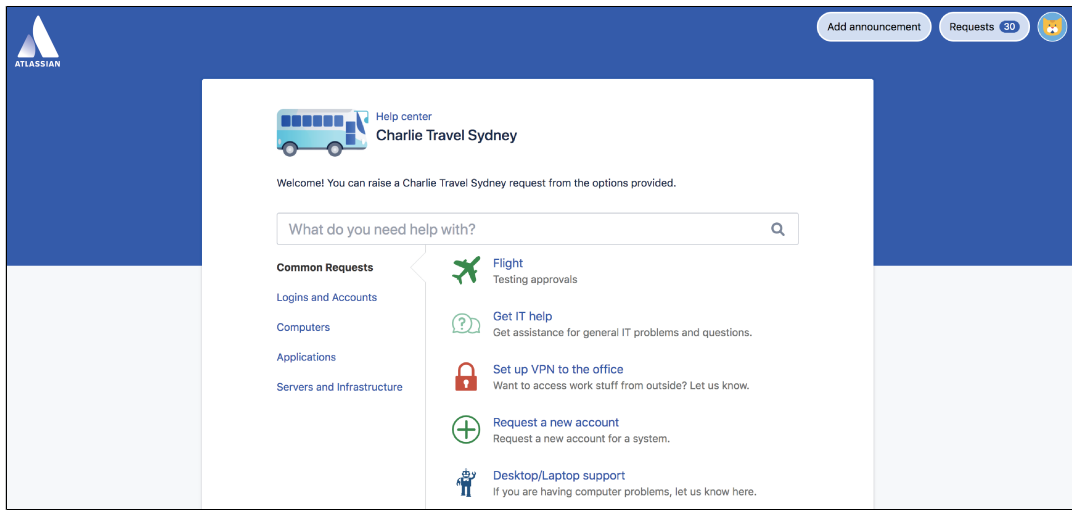


1. **Request name:** give the request an intuitive name by using keywords that your customers look for.
2. **Issue type:** add a new request type by selecting the issue type the request is based on.
3. **Description:** help your customers choose the right request type by providing helpful descriptions.
4. **Edit fields:** customize request fields and workflow statuses, and add request types to groups on your customer portal.

A single issue type can be the basis for many different request types. For example, the *Service Request* issue type serves as the basis for both the "Get IT help" and "Connect to wi-fi" requests.

Organize request types into groups

We recommend using groups if you have seven or more request types, so you can make your request types easier to find on the customer portal. You must have more than one group for the groups to appear in the customer portal. For example, 'Logins and Accounts', 'Applications', and 'Common Requests':

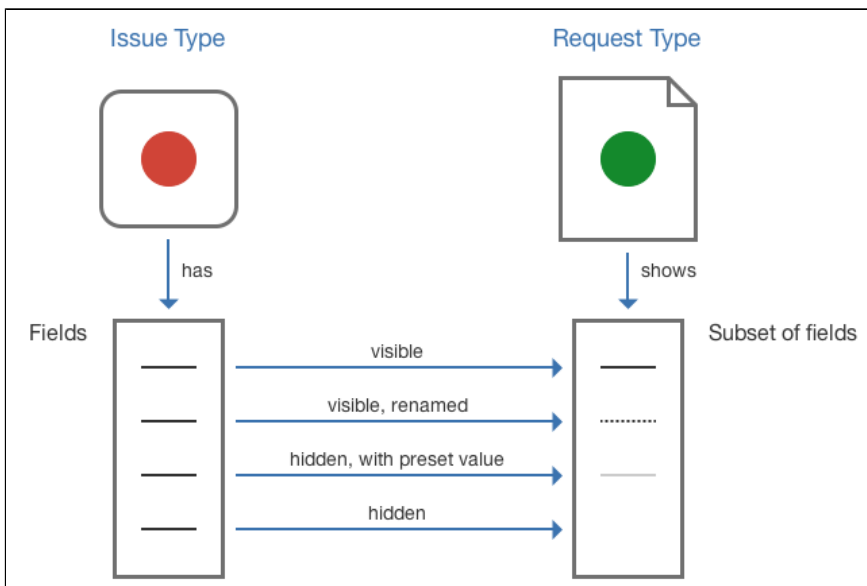


Administrators and project administrators can manage request type groups in **Project settings > Request types**. Click on a group to add new or existing request types to it. You can also create a new group by clicking **+ Add group** and these request type groups are unique to each service desk project.

- Drag and drop request types to rearrange them in your groups (and, consequently, on your customer portal).
- If you assign multiple groups to a single request type, the request type will appear on multiple tabs.

Customize the fields on a request type

The fields and descriptions that appear in a request type are based on the field configured for the issue type (that is, the issue type the request type is based on).



When editing the request type fields, you can use the **Fieldstab** to change the default Jira field names to more customer friendly language. For example, the "Summary" field appears as "What do you need?" for customers.

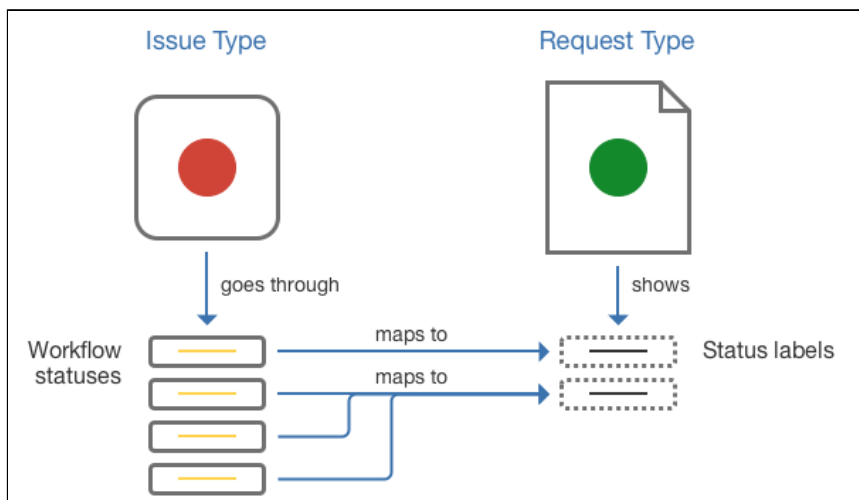
Display name	Required	Field help (Optional)	Actions
What do you need?	Yes	e.g. Wireless PC keyboard	Update Cancel
Issue field: Summary			
Why do you need this?	No		Hide Remove
How urgent is it?	No		Hide Remove

You can also keep fields hidden but available on the request type so that their value can be used for other processes. For more details about how different types of fields work in Jira Service Management, see [Hidden fields and unsupported fields](#).

If the issue type doesn't have the fields you need, you must [add a field](#) to the Jira issue type that the request type is based on. If the issue type uses multiple screen schemes, the new field must be available in the create screen. See [Associating a screen with an issue operation](#).

Customize the workflow statuses for a request type

Jira Service Management uses the workflow associated with the request's issue type for the flow of the request.



You can re-map the default workflow statuses to more customer friendly statuses that will appear for customers, and you can also map multiple statuses to a single customer status to simplify the appearance of the workflow. Use the **Workflow statuses** tab to customize the workflow that customers will see.

The screenshot shows the 'Workflow Statuses' configuration interface. It has two tabs: 'Fields' and 'Workflow Statuses'. The main heading is 'Workflow status in JIRA' with a 'view workflow' link. To the right is the heading 'Status name to show customer'. Below this, there are three rows of configuration:

Workflow Status in JIRA	Status name to show customer
Resolved	Completed
Waiting for customer	Waiting for customer
Waiting for approval	Waiting for approval

At the bottom of the configuration area, there are two buttons: 'Save' and 'Discard unsaved changes'.

Only changes between these customer-visible 'status names' will be reflected in the Customer Portal and its notifications (e.g. a transition between two workflow statuses can be hidden on the Portal by giving them the same 'status name'). For more information about notifications, see [Managing service project notifications](#).

If you need to change the workflow of a request, you must edit the workflow associated with the service project by going to **Project settings > Workflow**.

Hidden fields and unsupported fields

Each request type in a service project is based on an issue type. Every issue type has a set of allowed (and possibly required) fields associated with it. As you set up the request type, you can choose to include fields that are hidden on the customer portal but still provide a value to assist with your internal processes and reporting. For example, you might want to set the value of the "Label" field as "hardware" for the "Request new hardware" request type, and set the value as "software" for the "Request new software" request type.

Some fields used by an issue type are not supported for use in the customer portal; if you include these fields on a request type, they will automatically be added to the **Hidden fields with preset values** section and you'll be required to set a value for them.

Other fields aren't supported for use in Jira Service Management.

These fields can be added to the request type and given a preset value, but you cannot make them visible on the customer portal:

- Assignee
- Linked issues
- Any fields that are defined by other Jira applications
- Group, project, and version picker custom fields
- Security level

These types of fields can't be added to a request type and won't appear in the in the Add a field dialog:

- Issue type
- Log work
- Reporter
- Time tracking

Set up URLs with auto-populated request fields

You can generate URLs that will automatically populate selected request fields with contextual data. This way you can direct your customers to the customer portal from an external website and transfer certain details into the request fields.

- If an incorrect value is loaded via the URL, the field can display incorrect malformed data.
- The URL max length is 2048 characters.

- All values must be URI encoded.
- If the custom field is a default field from Jira or Jira Service Management, it will use its field name, e.g. Summary, Priority. If it's a user-generated field, it will require the custom field ID. [Learn how to find ID for custom fields](#)

URLs can fill in the following request field types:

Field type	Example usage	Description
Summary	<code>&summary=Hello%20World</code>	URI encode value and attach it to the custom field.
Description	<code>&description=About%20us.</code>	
Priority	<code>&priority=4</code>	
Text Textarea	<code>&customfield_1000=Lost%20phone</code>	
Date Due date	<code>&duedate=12/24/95</code>	Format = MM/DD/YY
Date time	<code>&customfield_1000=12/14/21-14:39:21</code>	Format = MM/DD/YY-HH:MM
Label picker	<code>&customfield_1000=hello,atlassian</code>	Comma separated labels.
Checkbox Multi-select	<code>&customfield_1000=500,501,502</code>	Pass values of the option ID to be selected, separated by comma.
Radio Select	<code>&customfield_1000=500</code>	Pass a single value to be selected.
Cascading select	<code>&customfield_1000=500,1</code>	Pass 1 or two values. The first value is the first select option to be selected. If a second value is passed, then it is the second select value to be selected.

Troubleshooting issues with request types

This page contains information about the errors and problems that you might have when setting up request types for your service project.

Issue	Resolution
<p>Cannot delete the request type because it is the default request type for the email channel.</p> <p>If you see this error when trying to delete a request type, it means that the email channel for your service project uses this request type as the default one for all the requests coming from emails. When Jira Service Management pulls emails from the associated email account and creates requests, this request type is assigned to the requests automatically.</p>	<p>Jira administrators can change the default request type for email requests to be another one by going to Project settings > Email settings in your service project. For more information about the email channel setup, see Receiving requests by email.</p>

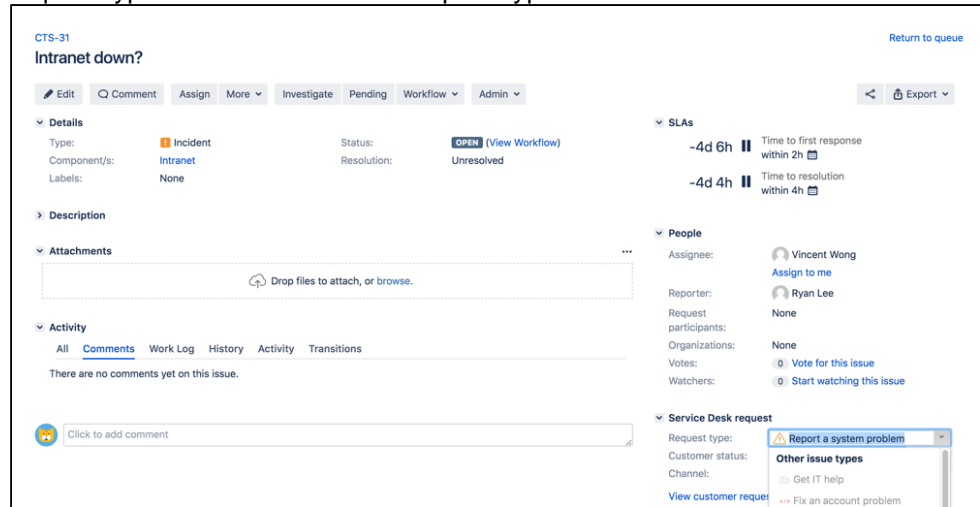
<p>Cannot show a hidden field or make an optional field required because the request type is the default for the email channel.</p> <p>When Jira Service Management creates new requests from emails sent to the email account associated with your service desk, it copies the email subject to the Summary field and the email content to the Description field. When more fields are required, Jira Service Management cannot parse emails to fill them in with correct values.</p>	<p>You have the following options:</p> <ul style="list-style-type: none">• If you want to show a hidden field, make it an optional one.• Ask your Jira administrator to change the default request type for the email channel to use a different request type, and then modify your request type to include more required fields. You can also create a new request type for the email channel if no existing types are suitable. For more information about the email channel setup, see Receiving requests by email.
---	---

Request type displayed as "No match" in agent view.

In Jira Service Management, the service project request type is stored with the PortalKey /RequestType Name value. For example, a "New Feature" request created in your "HelpDesk" Customer Portal would have the HelpDesk /NewFeature value. When you move this request to a new project, the HelpDesk /NewFeature value no longer matches the new project's Customer Portal name and request type values.

You have the following options.

- When you move a single issue to a new project, simply edit the service project request type field with the correct request type:



- If you need to move a group of issues, you can search for issues with the same issue type in your existing project and then use the Bulk Edit wizard. In the third step, check **Change Customer Request Type** and select the request type that applies to this group of issues.

Setting up SLAs

Good service is what keeps customers coming back and a key ingredient of good service is responsiveness.

With Jira Service Management, you can keep your team on track by setting goals for how quickly you manage customer issues. If these goals are set by your customer contracts, you might know them as Service Level Agreements, or SLAs.

SLAs track the progress of things like:

- Respond to all requests within 2 hours.
- Resolve high-priority requests within 24 hours.

How SLAs look

To see how SLAs are displayed in the agent view and customer portal in different scenarios, see [How teams see SLAs](#).


How to set SLAs

Your Jira admin or project admin can set SLAs in **Project settings > SLAs**.

When you set an SLA, you select three things:

- Who can view your SLA: only agents or also customers on the customer portal
- A time metric, which defines how and when time will be measured
- A goal, which defines the target to be met

Select who can view your SLA

 You need a Data Center license to use this feature.

By default, SLAs are visible only for agents. You can select to display each SLA individually also for customers on the customer portal. Customers will see the status of your SLA and its time metric, but they won't see the goal itself (for example, *within 5d*).

New SLA

Who can view this SLA
You can allow customers to view this SLA in the customer portal or keep it only for agents.

Show in customer portal

Yes
 No (default)

Time metric
Time will be measured between the **Start** and **Stop** conditions below.

Start* → **Pause on** → **Stop***

Begin counting time when Time is not counted during Finish counting time when

Set a time metric

The time metric works like a stopwatch, tracking the time between two points in an issue's lifecycle. For example, you might start time when an issue is created, pause time while you wait for the customer to respond, and stop time when the issue is resolved.

Jira Service Management has pre-configured time metrics to cover the most common IT requirements, but you can modify these or create your own as needed.

To create a new metric, from your service desk project sidebar, select **Project settings > SLAs > Create SLA** and fill in the following conditions:

Condition	Description
Name	This name (for example <i>Time to resolution</i>) will appear to agents in the SLAs section of issues. Agents should be able to read the name and know what they're being measured on. You can't change the name of an SLA metric after you've saved it.
Start	Time starts being counted against the SLA when any of these occur (for example, <i>Issue Created</i>).
Pause on	Time doesn't get counted against the SLA when at least one of these conditions apply to the issue (for example, <i>Status: Waiting for Customer</i>).
Stop	Time stops being counted against the SLA when any of these occur (for example, <i>Resolution : Set</i>).

You can set multiple conditions for the start, stop, and pause time. Here's an example of the conditions set for the *Time to resolution* SLA.

Time metric

Time will be measured between the **Start** and **Stop** conditions below.

Start

Begin counting time when

[Clear selected items](#)

- Issue Created
- Resolution: Cleared
- Issue Created
- Assignee: From Unassigned
- Assignee: To Unassigned
- Assignee: Changed
- Entered Status: Awaiting CAB a...
- Entered Status: Defined

→

Pause on

Time is not counted during

- Assignee: Set
- Assignee: Not Set
- Status: Awaiting CAB approval
- Status: Defined
- Status: Resolved
- Status: Waiting for customer
- Status: Peer review / change ...
- Status: Closed
- Status: Under Review

→

Stop

Finish counting time when

- Resolution: Set
- Assignee: From Unassigned
- Assignee: To Unassigned
- Assignee: Changed
- Entered Status: Awaiting CAB ...
- Entered Status: Defined
- Entered Status: Resolved
- Entered Status: Waiting for cu...
- Entered Status: Peer review / a...

i Check out [Example: creating an SLA with multiple cycles](#) to learn how to create a more complex SLA by starting and stopping the time counter throughout the workflow.

Set a goal

In the **Goals** section of the SLA metric, select the type of issue you want to track (**Issues**), how quickly you want to resolve it (**Goal**), and your teams working hours (**Calendar**).

Here are some examples of goals you might set:

- Resolve blocker issues within 24 hours.
- Resolve blocker issues created by the Build Engineering team within 12 hours.
- Resolve blocker issues created by the Accounting team within 36 hours.

In the **New SLA** or **Edit SLA** screen, fill out the following fields to define a goal:

Field	Description
Issues	You can enter specific issue criteria using Jira Query Language (JQL) . Base goals on criteria that remain relatively constant throughout an issue's lifecycle (for example, an issue's priority rather than its workflow status).
Goal	This is where you specify the target time for the SLA conditions you previously set. When specifying SLA goals that use a fraction of an hour, write the time as Xh Ym (for example, 3h 30m). You can write SLA goals as hours and minutes, but not days.
Calendar	The calendar allows you to specify working hours when time can be counted against SLAs.

You can drag goals in order of importance. An issue is tracked against the first goal criteria it matches on the list:

Goals

Issues will be checked against this list, top to bottom, and assigned a time target based on the first matching JQL statement.

Issues (JQL)	Goal	Calendar	
<input type="text" value=""/>	<input type="text" value=""/> (e.g. 4h 30m)	Default 24/7 calendar	Add
priority = Highest	36h	Default 24/7 calendar	Delete
priority = Blocker	24h	Default 24/7 calendar	Delete
All remaining issues	No target	Default 24/7 calendar	

Set your working hours

Jira Service Management lets you create calendars that match the working hours of your team including lunch breaks, holidays, and weekends. You can then assign that calendar to an SLA, so the clock stops when your team are away.

Read [Create and edit SLA calendars](#) to learn how to do this.

SLA calendars are unique to each service project. See [Example: Creating a basic SLA](#) for an example of setting up an SLA that uses a 9-5 working day SLA calendar.

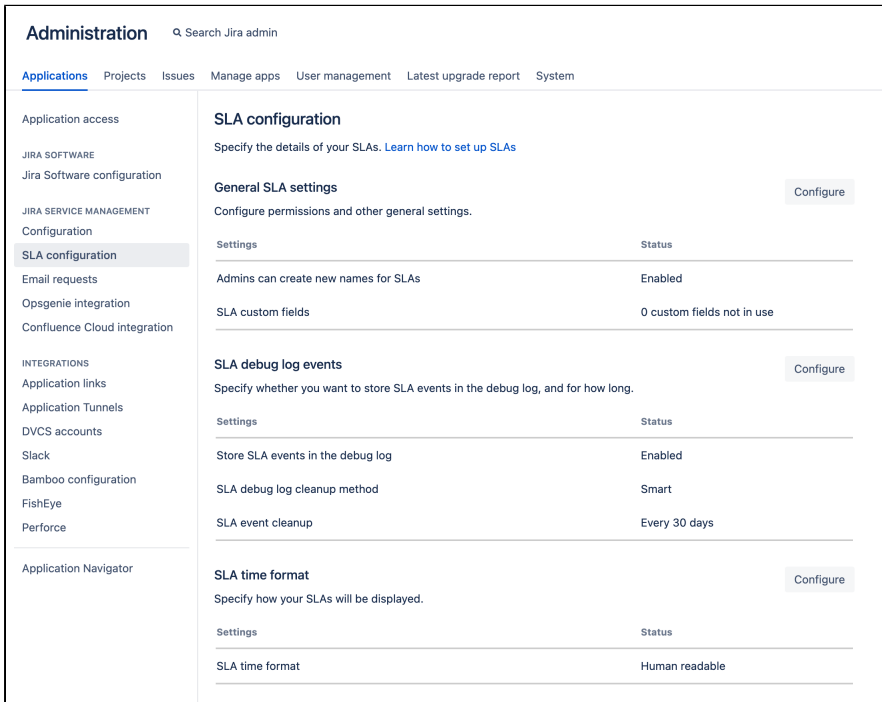
Configure your SLA settings

As a Jira administrator, you can manage permissions, formatting, and other SLA settings.

To configure your SLAs:

1. Go to **Administration > Applications**.

2. Select **SLA configuration**.



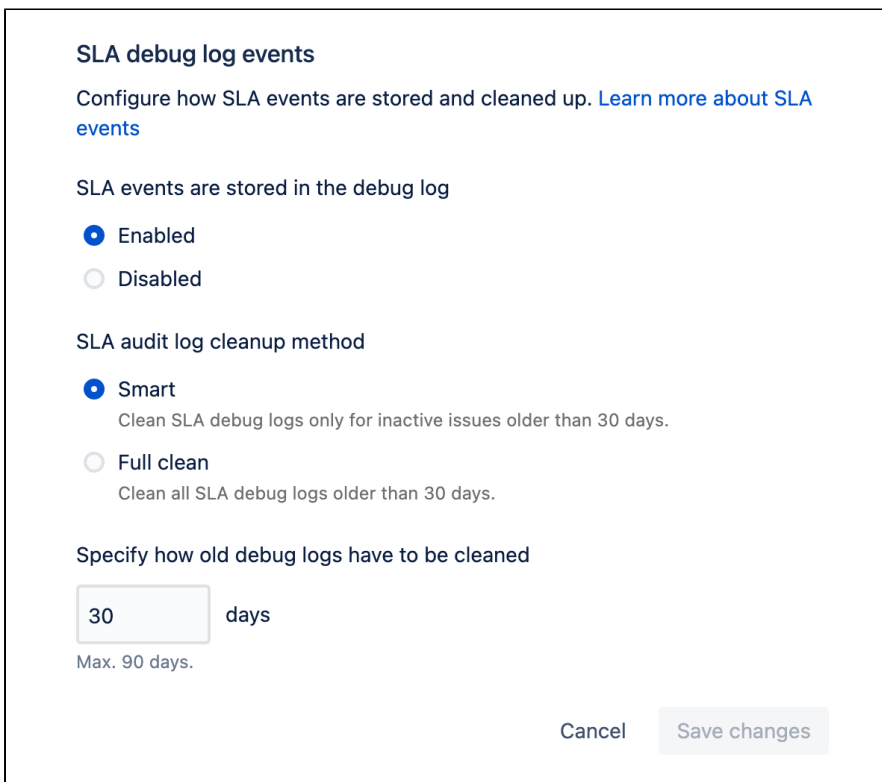
3. Find the section you want to manage.

4. Select **Configure**.

Clean up SLA debug log events

Whenever a Jira Service Management issue event is triggered or an SLA is created or deleted, Jira Service Management updates the corresponding SLAs. The SLA debug log feature creates an entry in the database (AO_54307E_SLAAUDITLOG and AO_54307E_SLAAUDITLOGDATA tables) to reflect that change.

As a Jira administrator, you can decide how you want the SLA debug log to be cleaned up. You can also select how often the events will be deleted from the database.



Configure thread processing

To increase the performance of Jira Service Management, we run SLA calculations in background tasks. A thread, in this context, executes those tasks, in parallel with other threads.

SLA thread configuration

Thread processing

Select your off-thread event processing mode. [Learn more about thread processing](#)

Processing mode

Multi-threaded serialized processing RECOMMENDED

Cluster efficient, recoverable, and resilient. Should be the default processor.

Set the maximum SLA thread count ⓘ

5

Select a number between 1 and 20

Multi-threaded in-memory processing

Legacy processing mode. Only revert to it temporarily if instructed.

No background processing

All processing will be done in the request thread, locking the response until it's finished. You might experience a significant delay to the response time.

Cancel Save changes

You can configure the processing method as well as the thread count for your SLA calculations.

Processing method	Description
Multi-threaded serialized processing (Number of threads is configurable)	<p>We recommend using this processing method.</p> <p>Introduced in version 4.21.0, this new processing method is self-healing, more cluster-efficient, and more reliable, specifically when Jira is running in a multi-node cluster mode.</p> <p>This method allows any Jira node to process any of the stored events. Unlike the in-memory processing mode, application restarts don't affect SLA processing with this method. Even if you only have one node in your cluster, SLA calculations will resume from the place they were interrupted should any interruption happen.</p> <p>It uses historical event data stored in the database to calculate SLAs in the order the events arrived for an issue. The thread pool in this method is configurable in the range of 1-20 threads.</p>
Multi-threaded in-memory processing (Number of threads is configurable)	<p>This method is considered a legacy processing mode. The thread pool here is configurable in the same way as in the serialized processing method in the range of 1-20 threads.</p> <p>In this mode, all processed events are bound to the node they had happened on. When the application stops on one node, all unprocessed events are lost, and not included in the SLA history until it's manually reconstructed (for versions before 5.0) or until another event happens on the same issue (from version 5.0).</p>


No background processing (Number of threads isn't configurable)	In this method, all processing is done in the request thread, locking the response (user feedback) until the SLA calculation is finished. If you select this processing mode, you might experience significant decrease in the perceived performance of actions like issue creation, comment creation, or issue transition.
---	---

SLA calculations result in querying the database to fetch information. Our databases use connection pools to limit simultaneous active connections to it. In case of no background processing, we don't control the pool count as it works on HTTP threads. This may lead to a fast database connection pool exhaustion under load. [Learn more about database connections](#)

Configure the SLA thread count

If you select the multi-threaded serialized or the multi-threaded in-memory processing method, you can configure the thread pool count for your SLA calculations in the range of 1-20. Thread count values are synchronized across all nodes without restarting the processors.

The default number of SLA threads per node is set to 5. However, in case the SLA calculations aren't keeping up with issue updates, we suggest using the recommended multi-threaded serialized processing method and bumping up the SLA thread count in small increments while observing the system response.

 Never set the SLA processing thread count to values larger than your database connection pool. Our recommendation is to not exceed 25% of your max database connection pool. To better understand the utilization and size of your database connection pool, check out the [Database Monitoring admin menu](#).

Updating SLAs


When you change the configuration of existing SLAs, we follow these rules:

- We don't recalculate, or touch in any way, cycles that have already been completed. A cycle is completed when a request fulfills the *Start* and *Stop* conditions (depending on your configuration and events, another cycle of the same SLA might then be started).
- We recalculate ongoing SLAs so they measure against your new configuration and metrics.

It might happen that ongoing SLAs will be completed or discarded entirely after changing the configuration. This can happen if you remove the previous *Start* or *Stop* conditions or change them to something else, which will trigger the completion. As mentioned earlier, ongoing SLAs will be recalculated to use your new configuration.


Recalculate your SLAs for a single issue

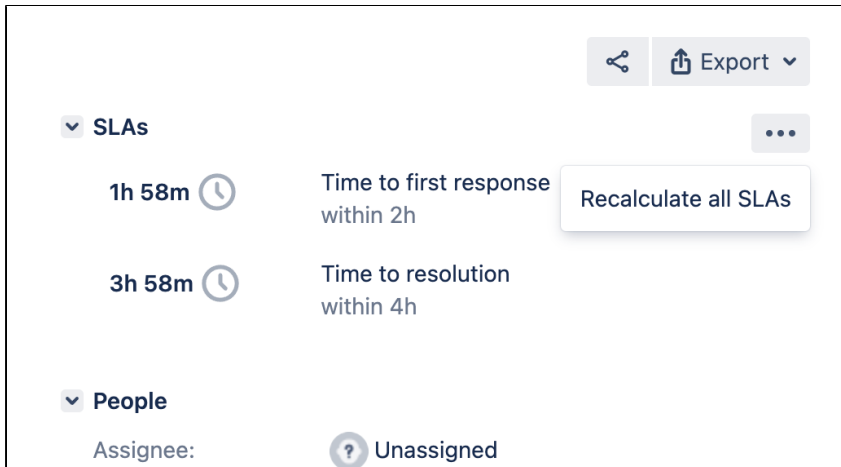
Occasionally, issue events that should trigger a change to an SLA value are missed or interpreted incorrectly resulting in an incorrect SLA value showing in the queues or the issue view. Usually, SLA calculations will catch up to missed events and correct the values, but if they don't, you can recalculate the SLAs for an issue in the issue view.

 SLA recalculation replaces the existing SLAs values using the current SLA configurations for the project. This operation is irreversible and may result in data loss.

To recalculate your SLAs for an issue:

1. Go to the issue view.

- From the SLAs section select the more actions menu  > **Recalculate all SLAs**.



- Select **Recalculate**.

Best practice SLA usage

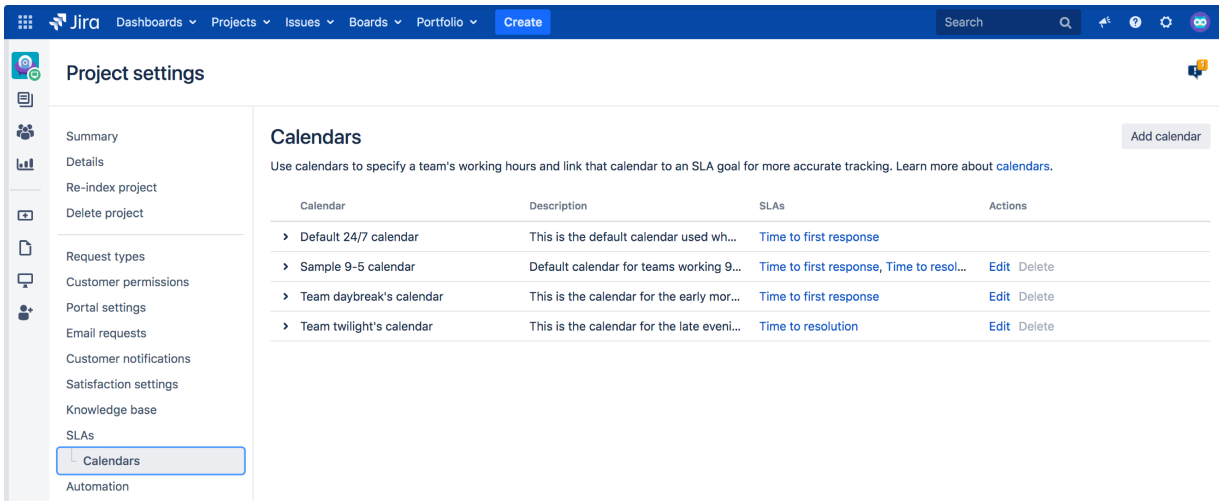
- Try to select an **Assignee** who's not the **Reporter** of an issue. If you assign the same user, your SLA might work incorrectly.
- If issue data changes in such a way that the goals for the issue change (for example, the priority changes from Critical to Blocker), the time against the previous goal will be tracked against the new goal, for open issues only.
For example, if the Support team spent an hour on a Critical issue, when the issue is escalated to Blocker, the hour still counts against the new goal, even if it causes the SLA to be breached.
- If you're using filters in your service project to track SLA data, specifically **priorities**, make sure these filters include all the priorities defined in the associated priority scheme. See [Associating priorities with projects](#) for more details.
- We do not recommend setting up a goal to be dependent on a different SLA.

Create and edit SLA calendars

When you create an SLA, you can link that SLA to a calendar to ensure the clock only ticks when your team are available. To support different ways of working, you can create calendars that take into account shift work, breaks, and holidays.

To get you started, Jira Service Management comes with two default calendars:

- Default 24/7 calendar. This is used when no calendar has been assigned to an SLA goal.
- Sample 9-5 calendar. This tracks hours from 9am to 5pm, Monday through Friday.



Create a new calendar

To create a new calendar in your service project:

1. Go to **Project settings > Calendars** (underneath **SLAs**).
2. In the top right corner, select **Add calendar**.
3. Fill out the **Name** and **Description**, so it's easy to know what hours are being tracked.
4. Choose the working hours for each day, and click '+' to add a new row.
5. Select **Save**.

Working hours

<input checked="" type="checkbox"/> Monday	09:00	to	12:00	+	
	13:00	to	17:00	-	
	Total: 7h				
<input checked="" type="checkbox"/> Tuesday	08:00	to	10:00	+	
	10:30	to	14:00	-	
	15:00	to	18:00	-	
	Total: 8h 30m				
<input checked="" type="checkbox"/> Wednesday	10:00	to	17:00	+	⚠
	09:00	to	17:00		
	Total: 8h				
<input checked="" type="checkbox"/> Thursday	09:00	to	17:00	+	
	Total: 8h				
<input checked="" type="checkbox"/> Friday	09:00	to	17:00	+	
	Total: 8h				

This interval overlaps with at least one other interval. Check your working hours before saving.

✔ Look at the **Total** count for each day to make sure you're tracking the correct number of hours.

Edit or delete a calendar

To edit an existing calendar in your service project:

1. Go to **Project settings > Calendars** (underneath **SLAs**).
2. Find the calendar, select either **Edit** or **Delete**.
 - If deleting, you can't delete the **Default 24/7 calendar**. You also need to remove a calendar that's being used by an SLA from the SLA itself, before you can delete it here.
 - If editing, you can't edit the **Default 24/7 calendar**.
3. Once you've made your changes, select **Save**.

When you make changes, we'll recalculate any project SLAs linked to this calendar.


Import holidays

To import holidays to a calendar:

1. Go to the **Holidays** section at the bottom of the calendar.
2. Select the **Import holidays** link.
3. Choose your ICS file, and set it to **UTC/GMT+10**.
4. Select **Import**. You'll see a modal that lists the holidays about to be imported. Delete unwanted holidays by clicking 'x'.
5. Select **Import**.

Imported holidays will be added to any existing holidays in the calendar. If you'd like a holiday to repeat the following year, check the **Repeat yearly** box and you won't have to import it again.

Holidays
Add or import holidays to your calendar so non-working days don't count towards your SLAs.


Add holidays  [Import holidays](#)

Future holidays [2019](#) [2020](#)

Date	Holiday	Repeat yearly	
14/Aug/19	Australia: Ekka People's Day (Regional)	<input type="checkbox"/>	Delete
24/Dec/19	Australia: Christmas Eve (Regional)	<input type="checkbox"/>	Delete
25/Dec/19	Australia: Christmas Day	<input checked="" type="checkbox"/>	Delete
26/Dec/19	Australia: Boxing Day (Regional)	<input checked="" type="checkbox"/>	Delete
31/Dec/19	Australia: New Year's Eve (Regional)	<input type="checkbox"/>	Delete

Past holidays [2019](#) [2018](#) [2017](#) [2016](#) [2015](#) [2014](#) [2013](#) [2012](#) [2011](#) [2010](#) [2009](#) [2008](#)

Date	Holiday	Repeat yearly	
01/Jan/19	Australia: New Year's Day	<input type="checkbox"/>	Delete
19/Apr/19	Australia: Good Friday	<input type="checkbox"/>	Delete
22/Apr/19	Australia: Easter Monday	<input type="checkbox"/>	Delete

 If you want to delete a recurring holiday you need to do this in the year it was created.

Link calendars to SLAs

To link a calendar to an SLA:

1. Go to **Project settings > SLAs**.
2. Choose the SLA you want to link to, select **Edit SLA**.
3. Set your **Start**, **Pause**, and **Stop** conditions.
4. In the **Goals** section under **Issues**, choose the type of issues you want to track.
5. Under **Goal**, choose how quickly you want to resolve them.
6. Under **Calendar**, select the edit icon and choose a calendar from the dropdown.
7. Select **Save**.

Goals

Issues will be checked against this list, top to bottom, and assigned a time target based on the first matching JQL statement.

Issues (JQL)	Goal	Calendar
<input type="text"/>	<input type="text"/> (e.g. 4h 30m)	Default 24/7 calendar <input type="button" value="Add"/>
⋮ <input type="text" value="priority = Blocker"/>	<input type="text" value="2h"/> (e.g. 4h 30m)	<div style="border: 1px solid #ccc; padding: 5px;"> Default 24/7 calendar <input checked="" type="checkbox"/> Sample 9-5 calendar Team daybreak's calendar Team twilight's calendar </div> <input type="button" value="Update"/> <input type="button" value="Cancel"/>
All remaining issues	8h	Team daybreak's calendar

To learn all about SLAs and how to set them up, go to [Setting up SLAs](#).

How teams see SLAs

Support teams rely on SLAs to plan and track their time against issues.

Jira Service Management shows SLAs in a human readable time, instead of only hours and minutes. That way, service project agents don't spend precious time working out what 78:00 means in days.

On this page:

- [How it looks](#)
- [How the clock works](#)
- [Multiple targets](#)
- [Sorting](#)

How it looks

The SLAs look a bit differently for agents and for customers, if they're enabled on the customer portal:

- **Agents** see a read-only version of the tab, so they know how the is configured. In the issue view, the SLAs section lists more information about the that the issue is being measured against.
- **Customers** see SLAs that have been enabled on the customer portal, but they only see the status, without the actual goal. They can still see if an is at risk or breached.

SLAs

- 3min Time to resolution within 5min
- 21min Time to assign within 30min
- 2h 51m Time to first response within 3h







SLAs

- 3min Time to resolution
- 21min Time to assign
- 2h 51m Time to first response

Ongoing SLAs



The tracker uses colors to indicate the urgency of an for an issue, based on the time remaining.

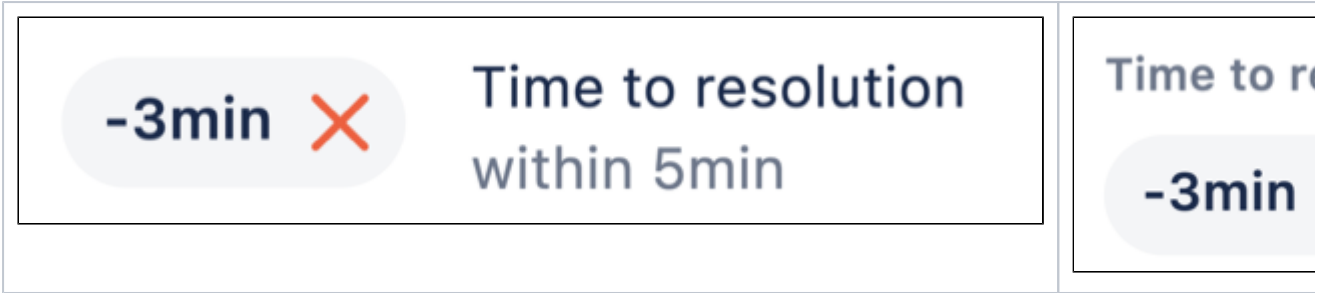
Agent view	Customer portal
<p>1h 1m Time to resolution within 1h 10m</p>	<p>Time to r 1h 1m </p>
<p>59min Time to resolution within 1h</p>	<p>Time to r 59min </p>

 <p>Time to resolution within 30min</p>	<p>Time to r</p> 
 <p>Time to resolution within 5min</p>	<p>Time to r</p> 
<p>1h </p> <p>Time to resolution within 1h</p>	<p>Time to r</p> <p>1h </p>

Completed SLAs

A completed displays the time remaining when the was completed (or the amount of time breached) and an icon to indicate whether the was completed successfully or unsuccessfully.

Agent view	Customer portal
 <p>Time to resolution within 1h</p>	<p>Time to r</p> 



Past cycles

SLAs can be configured to start, pause, and stop based on various conditions. After an SLA has stopped and if one of the start conditions is detected again, the SLA will be reset to start a new cycle.

The past cycle will be displayed in the **SLAs** panel in the issue view, underneath the SLAs name. You'll see an expander under that where the corresponding cycles are listed.

There will be as many previous cycles as times an SLA is reset.

i Past SLA cycles show only in the agent view and are not available in the customer view.

Here's how past SLA cycles are displayed.

Agent view	<div style="border: 1px solid #ccc; padding: 10px;"> <p>▼ SLAs</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="background-color: #f7941d; border-radius: 15px; padding: 5px 10px; display: flex; align-items: center; justify-content: center;"> 28min 🕒 </div> <div style="margin-left: 10px;"> <p>Time to response within 30min</p> <p>2 cycles ^</p> </div> </div> <div style="display: flex; flex-direction: column; align-items: center; margin-top: 5px;"> <div style="display: flex; align-items: center; margin-bottom: 5px;"> × 35m </div> <div style="display: flex; align-items: center;"> ✓ 3m </div> </div> </div>	<p>The SLA has two past cycles:</p> <ol style="list-style-type: none"> 1. The first cycle where the SLA wasn't breached, and 3 minutes passed. 2. The second cycle where the SLA was breached, and 35 minutes passed.
Customer view	<div style="border: 1px solid #ccc; padding: 10px; text-align: center;"> <h2 style="margin: 0;">SLAs</h2> <h3 style="margin: 0;">Time to response</h3> <div style="display: flex; align-items: center; justify-content: center; margin-top: 20px;"> <div style="background-color: #f7941d; border-radius: 15px; padding: 10px 20px; display: flex; align-items: center; justify-content: center;"> 28min 🕒 </div> </div> </div>	

Units of time

The following units represent time in an:

Seconds	n/a
Minutes	m (min if single unit)
Hours	h
Days	d
Weeks	w
Months	mo
Years	yr

Only 1 or 2 units, at a time, are displayed for simplicity, with a combination of the following:

min	
h	m
d	h
w	d
mo	
yr	

How the clock works

Jira Service Management calculates minutes, hours, days, and weeks by using the working hours set in the associated calendar. It calculates a month and a year by using approximations of 4 weeks and 12 months respectively.

Read [Create and edit calendars](#) to learn how to track your teams' working hours and assign calendars to an goal.

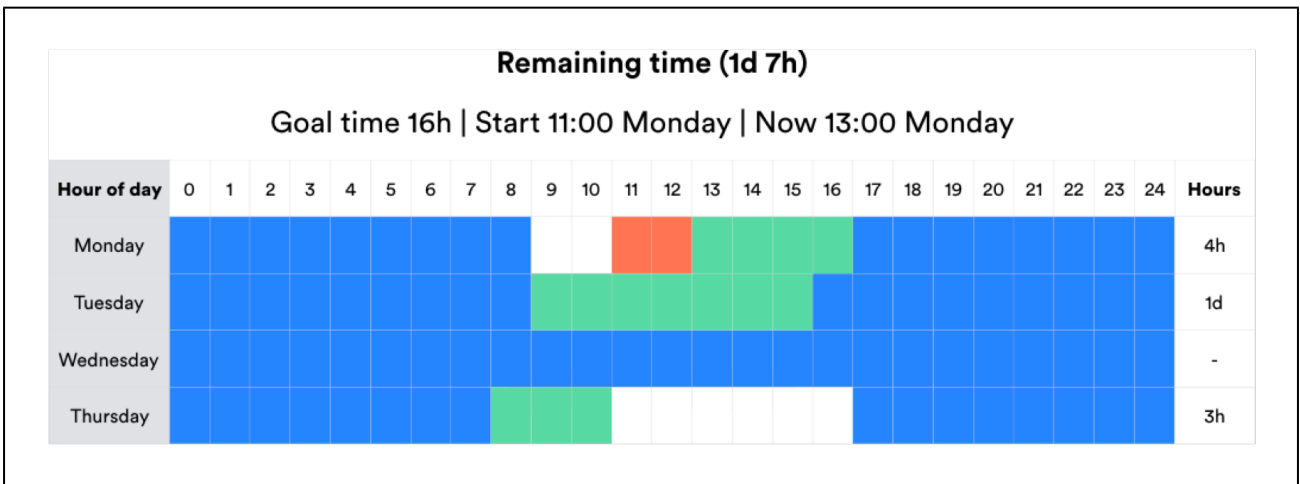
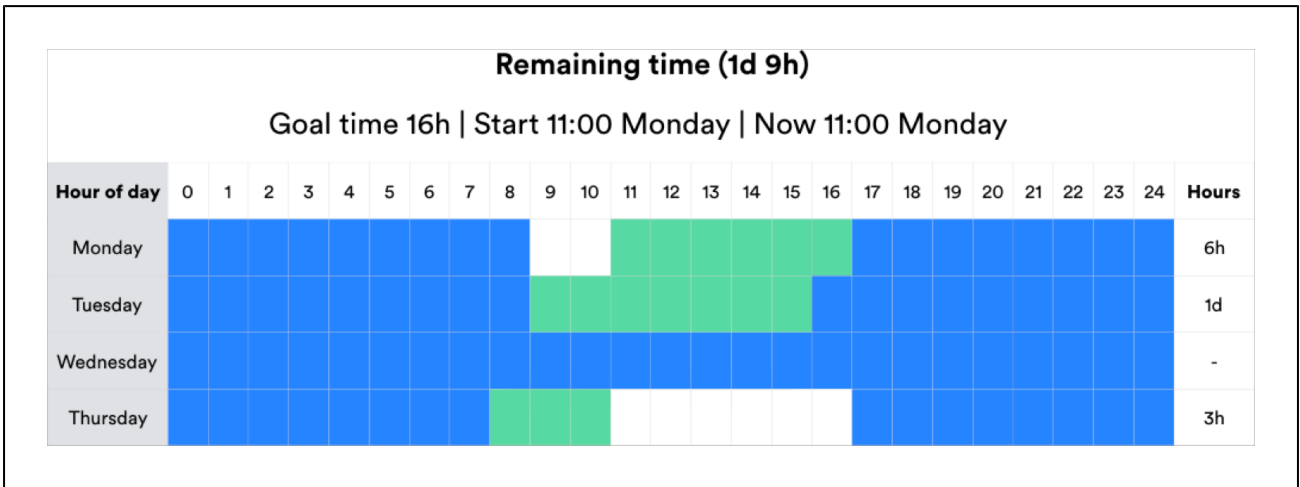
Remaining time

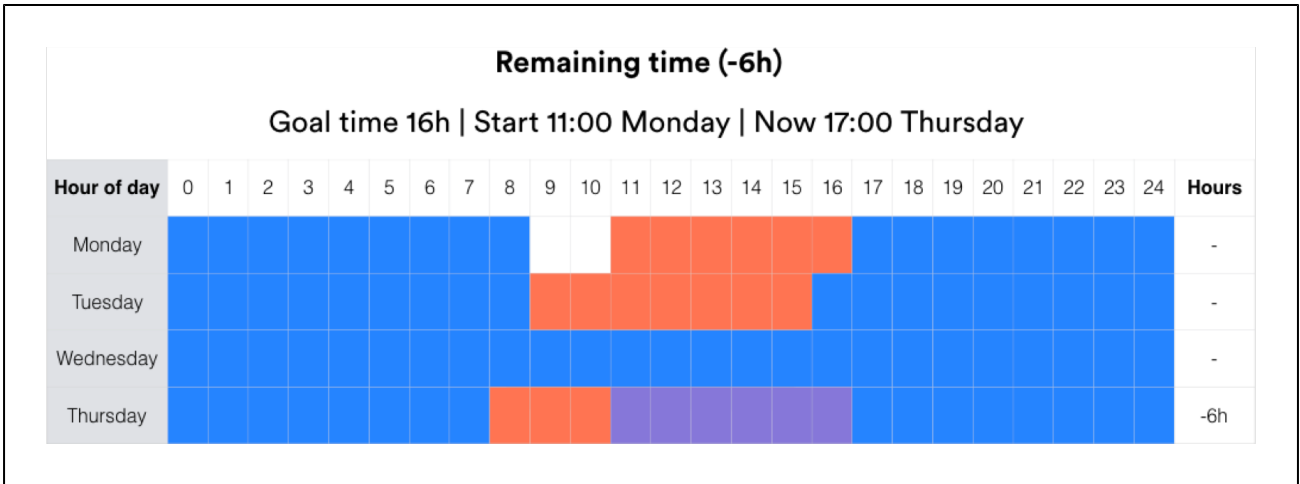
Remaining time is calculated by looking at the time from **now** until **due**, taking into account any pauses.

For example, Request A has an goal time of **16 hours**, with a start time of **11:00** on Monday, and working hours as follows:

- Monday 09:00-17:00
- Tuesday 09:00-16:00
- Wednesday not working
- Thursday 08:00-17:00
- Friday not working
- Saturday not working
- Sunday not working

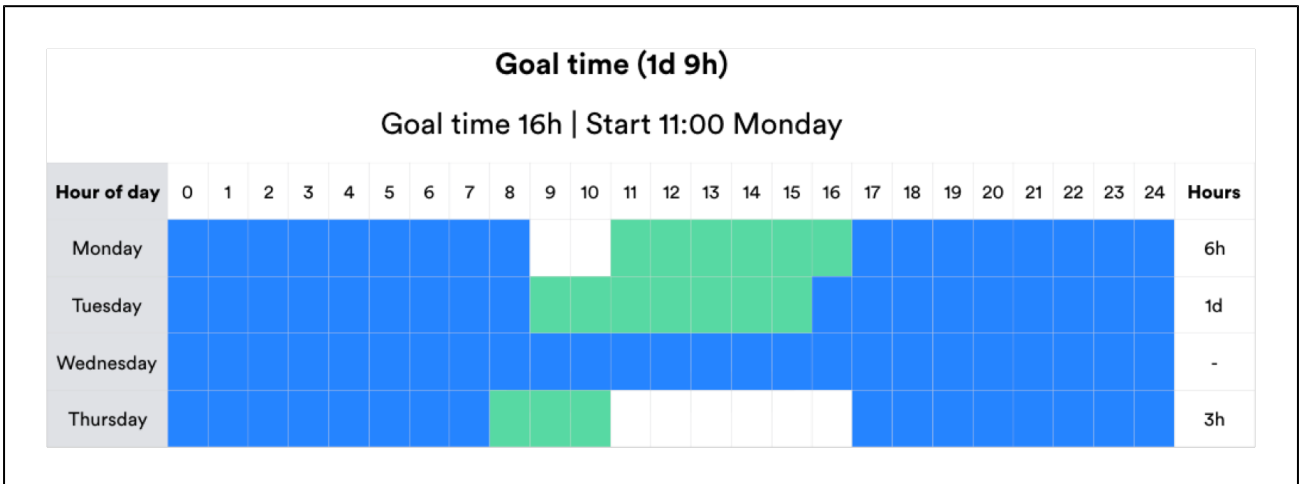
How to read the diagrams:





Goal time

Jira Service Management calculates goal time similar to remaining time, but **now** is set to **start** of goal. Goal time doesn't take any future pauses into account.



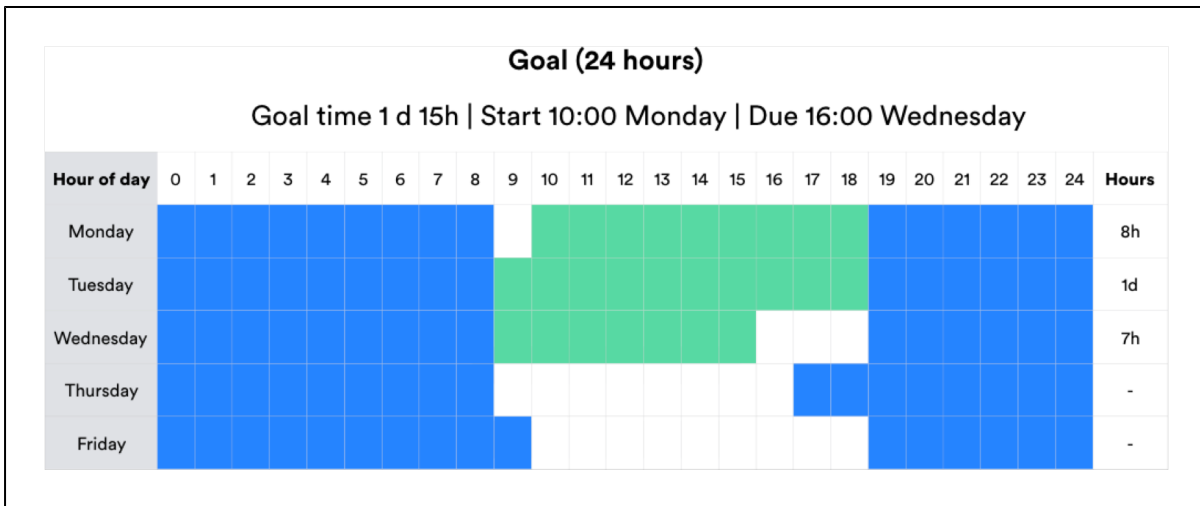
Regardless of the current time, the goal time is always calculated from the start time.

Partial days

In some situations, your Agents might see time displayed as something like 1d 26h. This happens when an starts and stops for a period of less than a day (for example, 7 hours in an 8 hour working day).

All partial days are accumulated as hours and minutes, but not converted to 1 day, because **the definition of a day will change depending on the working hours set in the calendar.**

For example, Request B has a goal time of 24 hours, with a start time of 10:00 on Monday, and working hours displayed in the diagram below. The goal is **1d 15h** and the is due at 16:00 on Wednesday.



It works like this:

1. From 10:00-18:00 on Monday is not a complete working day (according to the calendar). Total = 8 hours.
 2. From 09:00-18:00 on Tuesday is a complete working day (according to the calendar). Total = 1 day.
 3. From 09:00-16:00 on Wednesday (due time) is not a complete working day (according to the calendar). Total = 7 hours.
- 8 hours + 1 day + 7 hours = **1d 15h**

Multiple targets

If the issue meets the criteria for multiple SLAs, trackers for each will appear. In addition, if the has had multiple cycles, you can hover over the symbols for more details on how the was met for that particular cycle. For example, in an that is measured based on when an issue is *Waiting for support*, you can see whether the was met each time the issue started *Waiting for support*.

Sorting

When you view a list of issues (in a queue or elsewhere), you can sort them by their remaining times. Ongoing issues are listed first, with the shortest time remaining at the beginning of the list. Completed issues are ranked last, but aren't sorted by the remaining time.

Reporting on SLAs

The following describes reports we recommend for teams that track SLA goals which metrics we think are useful and why, and how to create reports.

To view your service project's reports or create new ones, select **Reports** in your service project sidebar. *You must be an administrator to create or edit reports.*



- If your SLA goals use overlapping JQL filters, your custom SLA reports may be inaccurate. [Read more about setting up SLA goals.](#)
- If your JQL filters use priorities, make sure these filters include all the priorities defined in the associated priority scheme. See [Associating priorities with projects](#) for more details.

On this page

- [Track success with percent of SLAs met](#)
- [Get insight into an agent's performance per SLA](#)

Track success with percent of SLAs met

By default, Jira Service Management includes an SLA goals report that shows how your team performed against its goals during the past week. This report shows the big picture.

But, if you're interested in trends, want to see a different timeframe or are just more of a visual person, you may find creating a custom report more useful.

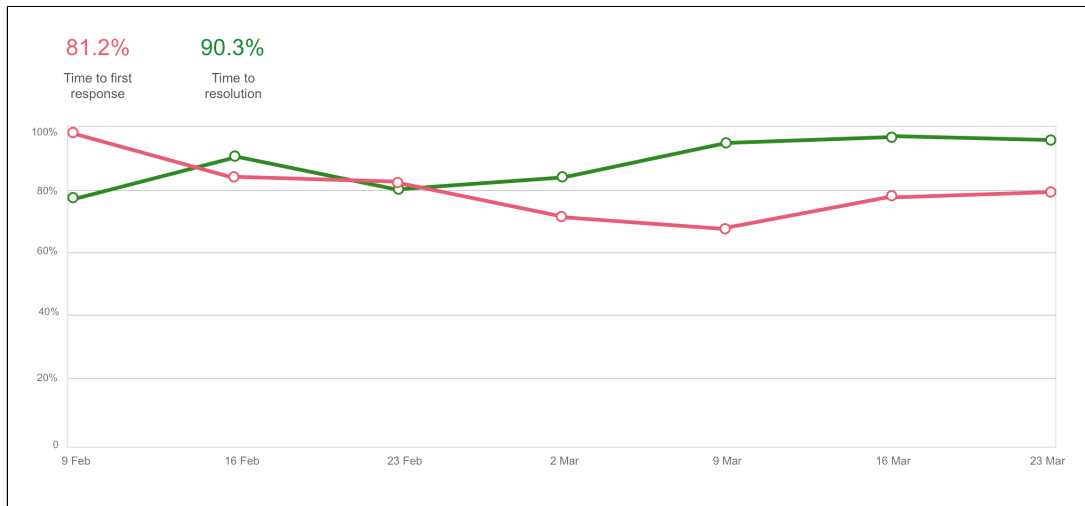
For example, you can create reports to see how well your team performs per request type. Create a report with the following series to drill into how your team is doing on each of your SLA goals for IT help requests:

- **Series**= Time to first response % met
- **Label**= Time to first response
- **Filter by**(advanced) = "Customer Request Type" = "Get IT help"

- **Series**= Time to resolution % met
- **Label**= Time to resolution
- **Filter by**(advanced) = "Customer Request Type" = "Get IT help"

You may find your team is speeding up on responding to customers with these requests but slowing down on resolving these issues. Looking at the details, you might see issues that are breaching your SLA goals feature words like "wi-fi" or "access".

Maybe you need to consider more reliable network hardware or you may need to train your organization about how to properly use the network. You may consider creating a knowledge base article that you can use to quickly answer these questions and increase the percentage of SLA goals met for resolving these requests.



✔ Select a data point on the chart and view all of the issues that were used to calculate the % met metric. We calculate the total from issues that successfully met the goal and those that breached the goal, regardless if they are paused or ongoing.

Get insight into an agent's performance per SLA

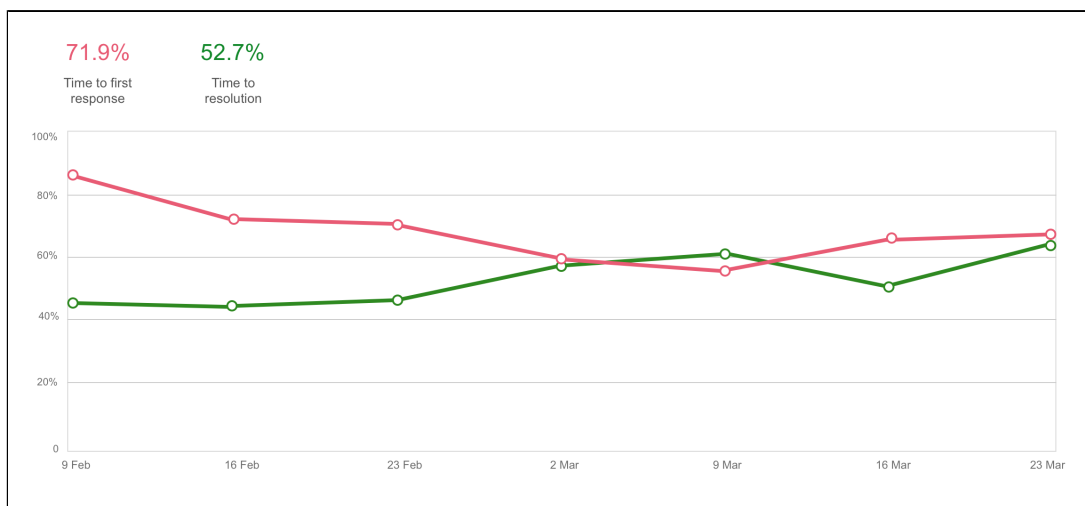
Line managers and other stakeholders may wish to view individual agent performance trends to help determine when agents are being stretched too thin or ensure work is being distributed appropriately.

Use the assignee field and make some reports for your agents to see how they are keeping up with their workload and where their strengths lie.

- **Series**= Time to first response % met
- **Label**= Time to first response
- **Filter by**(advanced) =assignee = agentUserName

- **Series**= Time to resolution % met
- **Label**= Time to resolution
- **Filter by**(advanced) =assignee = agentUserName

You may find that an agent is slow to respond but very quick to resolve requests. Working with the agent, you may be able to strategize how to better notify them about requests when they are raised.



Importing SLAs

You can reuse SLAs that exist in another service project, in your own project. Importing SLAs will import the full configuration of your SLA including calendars, and allow you to use the same SLAs in your project.

SLAs are not linked, so if you make a change to them in the source project, you'd need to reimport the SLAs to see them in your new project.

SLAs

Time to first response

Time to resolution

+ Create SLA

+ Import SLA configu...

How to import an SLA configuration

Before you import your SLAs we recommend you take a look at how the SLAs are set up in the source project. Make sure you have the required elements in your project. For example, you'll need to have the same **Start**, **Pause**, and **Stop** conditions available, and the same request types available. These elements are what's used to calculate your SLA.

To import an SLA configuration:

1. Go to **Project settings** in the sidebar.
2. Select **SLAs**.
3. Select **+ Import SLA configuration**.
4. Choose the project you want to import the SLAs from. If there's any errors or warnings that relate to your import, like duplicate SLA names, you'll be informed. You can fix some of these at a later date, or you can choose to stop the import, make the changes, and then reinitiate the import.
5. If you want to replace all existing SLAs, check the relevant box.
6. Select **Import**.

You've just imported the SLA and they're ready to go into your project. Note that these SLAs will not affect any closed issues, but if you have open issues their SLAs will be recalculated.

How to fix your SLA import

When you import SLAs you'll be notified of any issues with your import that you'll need to fix. Here's a list of the issues and resolutions to help you get your project up and running, as soon as possible.

Jump to issue...
<ul style="list-style-type: none">• Goal triggers• Goals• Calendars• Reports• Request and issue types

Issues with goal triggers

The start, pause, and/or stop conditions used by the SLAs you're importing don't exist in the project you're importing them too.

How to resolve

Depending on your project and what you need to use the SLAs for, you might want to:

- delete the condition from your imported SLA
- create the condition/s that's missing in your imported SLA.

Issues with goals

The start, pause, and/or stop conditions used by the SLAs you're importing don't exist in the project you're importing them too.

How to resolve

Edit the SLA's search filter and remove the incorrect criteria to fix it.

Issues with calendars

A calendar you're importing has no hours assigned to the working week.

How to resolve

Add working hours to the calendar, or delete the existing calendar and create another calendar. Learn more at [Create and edit SLA calendars](#).

Issues with reports

There's an existing report that uses an SLA you're about to delete during your import.

How to resolve

Fix the report(s) affected and either:

- replace the SLA in the affected report
- delete the report.

Issues with request and issue types

The request and/or issue types used by the SLAs you're importing don't exist in the project you're importing too.

How to resolve

Either delete references to these in the SLAs, or create the missing request types/issue types and add them to the JQL or condition.

Using JQL queries specific to SLAs

Jira Service Management includes specific JQL syntax that can help you sort through the details of your requests and issues, and make sure you're hitting your SLA goals. [Read more about JQL syntax.](#)



- If your SLA goals use overlapping JQL filters, your JQL queries may return unexpected results. [Read more about setting up SLA goals.](#)
- If your JQL filters use priorities, make sure these filters include all the priorities defined in the associated priority scheme. See [Associating priorities with projects](#) for more details.

On this page

- [Find issues breaching your SLA goals](#)
- [Find issues that have paused, completed and are still running an SLA clock](#)
- [Find issues based on their SLA clock](#)

Find issues breaching your SLA goals

There are two functions you can use to search for issues that are in a certain state of SLA goal-ness:

- **breached()** filters out issues whose last SLA cycle has failed to meet its target goal
- **everBreached()** filters out issues that have failed to meet their target goal

For example, if you wanted to find all the issues in your project that have successfully completed your first-response goals, use the following query:

```
"Time to first response" != everBreached()
```

Find issues that have paused, completed and are still running an SLA clock

There are three functions you can use to search for issues that are in a certain state of completion:

- **paused()** filters issues whose current SLA cycle is paused. This is determined by your SLA conditions. For example, you may pause an issue's SLA clock when the issue's status is set to "waiting for customer".
- **completed()** filters issues whose SLA cycle is complete, meaning they've reached one of their stop events.
- **running()** filters issues whose current SLA clock is running, meaning they haven't yet reached one of their stop events.
- **withincalendarhours()** filters issues whose SLA clock is running or not running according to the SLA calendar, *not* conditions.

The paused and running functions do not return issues whose SLA cycles haven't started yet.

For example, if you want to find all the issues that are paused while completing a time to resolution SLA cycle, use the following query:

```
"Time to resolution" = paused()
```

Find issues based on their SLA clock

There are two functions you can use to search for issues that have a certain amount of time on their SLA cycle's clock:

- **elapsed()** filters issues whose SLA cycle's clock meets a specified time condition since the ongoing SLA cycle's start event
- **remaining()** filters issues whose SLA cycle's clock meets a specified time condition before the issue will breach an SLA goal

For example, if you want to find requests that have been waiting for a first response for less than 10 min, use the following query:

```
"Time to first response" < elapsed("10m")
```

Or, if you want to find issues that will breach their resolution target within the next two hours, use this query:

```
"Time to resolution" < remaining ("2h")
```


Example: creating a basic SLA

This example looks at how you might create a very basic SLA for a service project with a basic workflow:

Basic SLA configuration

All highest and blocker issues must be resolved within 24 hours. You provide 24/7 support for certain customers (these issues are labeled with "24H"). You provide 9-5 support for all other customers, but you don't track SLA metrics for them.

Time to resolution

Time will be measured between the **Start** and **Stop** conditions below.

Start → **Pause on (Optional)** → **Stop**

Begin counting time when → Time is not counted during → Finish counting time when

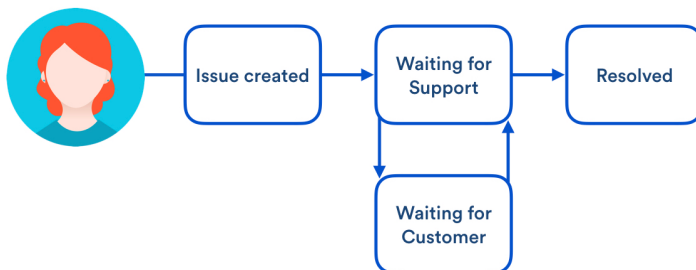
Issue Created → [] → Resolution: Set

Goals

Issues will be checked against this list, top to bottom, and assigned a time target based on the first matching JQL statement.

Issues (JQL)	Goal	Calendar
priority = Blocker OR priority = Highest AND labels = 24h	24h	Default 24/7 calendar
priority = Blocker OR priority = Highest	24h	Sample 9-5 Calendar
All remaining issues	No target	Sample 9-5 Calendar

Basic issue workflow



Example: creating an SLA that doesn't track continuous time

This example looks at how you might create a more complex SLA by pausing the time counter during the workflow:

Example SLA configuration

Support wants to complete all issues within 40 hours. Time spent waiting on the customer doesn't count against the 40 hour goal.

Time for support on all issues

Edit SLA Delete SLA

Time will be measured between the **Start** and **Stop** conditions below.

Start → **Pause on (Optional)** → **Stop**

Begin counting time when → Time is not counted during → Finish counting time when

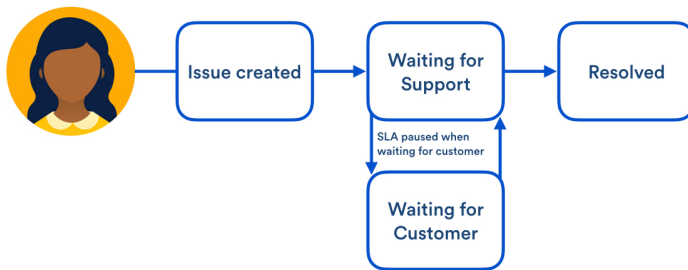
Issue Created → Status: Waiting for customer → Resolution: Set

Goals

Issues will be checked against this list, top to bottom, and assigned a time target based on the first matching JQL statement.

Issues (JQL)	Goal	Calendar
All remaining issues	40h	Default 24/7 calendar

Example issue workflow



Example: creating an SLA with multiple cycles

This example looks at how you might create a more complex SLA by starting and stopping the time counter throughout the workflow. You might set up an SLA like this to track response times (for example, how long it takes your team to respond each time a customer updates an issue with more information). This example also illustrates how goals for different issue criteria can be tracked from a single SLA.

Example SLA configuration

Support wants to respond to **Service requests** within two hours: this includes responding within two hours when the issue is created, as well as each time the issue is updated with more information from the customer.

All other issues have a response time goal of 24 hours.

Time to respond

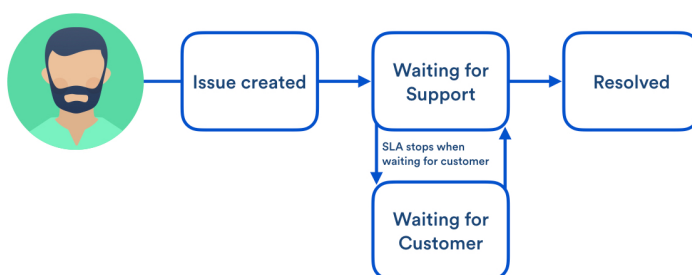
Time will be measured between the **Start** and **Stop** conditions below.

Start Begin counting time when	→	Pause on (Optional) Time is not counted during	→	Stop Finish counting time when
Entered Status: Waiting for support				Entered Status: Waiting for customer
Issue Created				Resolution: Set

Goals
Issues will be checked against this list, top to bottom, and assigned a time target based on the first matching JQL statement.

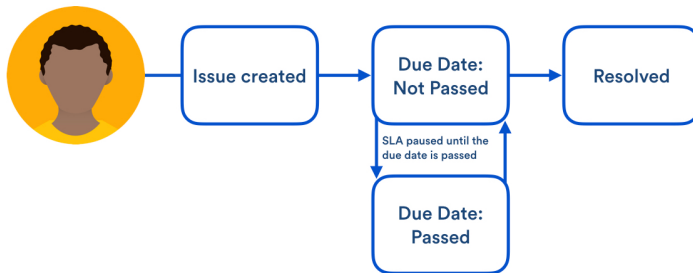
Issues (JQL)	Goal	Calendar
issuetype = "Service Request"	2h	Default 24/7 calendar
All remaining issues	24h	Default 24/7 calendar

Example issue workflow



✔ For further information about how SLAs with multiple start and stop conditions appear in the SLA tracker, see [Setting up SLAs](#).

Example: creating an SLA based on due date



Here's an example of how you might create a more dynamic SLA by pausing the time counter until a specified due date has passed. Set up an SLA like this if your team can't begin their work until a date in the future. For example, setting up a workstation when a new hire starts.

For this SLA to trigger, configure the **Due** field to display on the Issue type screen, and set the **Due** field when the issue gets created. Read [Defining issue type field values](#) to learn how to set this up.

Example SLA configuration

Support want to complete all hardware requests within 24 hours. They want to pause the SLA until they receive the hardware from their supplier (on the expected due date), then unpause the SLA when the due date has passed. All other issues have the same response time goal of 24 hours.

See the image below for how you could set up this SLA in Jira Service Management:

Time to close after due date passed Edit SLA Delete SLA

Time will be measured between the **Start** and **Stop** conditions below.

Start → **Pause on (Optional)** → **Stop**

Begin counting time when → Time is not counted during → Finish counting time when

Due Date: From Empty → Due Date: Not Passed → Resolution: Set

Issue Created


Goals

Issues will be checked against this list, top to bottom, and assigned a time target based on the first matching JQL statement.

Issues (JQL)	Goal	Calendar
"Customer Request Type" = "Request new hardware (CTF)"	24h	Default 24/7 calendar
All remaining issues	24h	Default 24/7 calendar

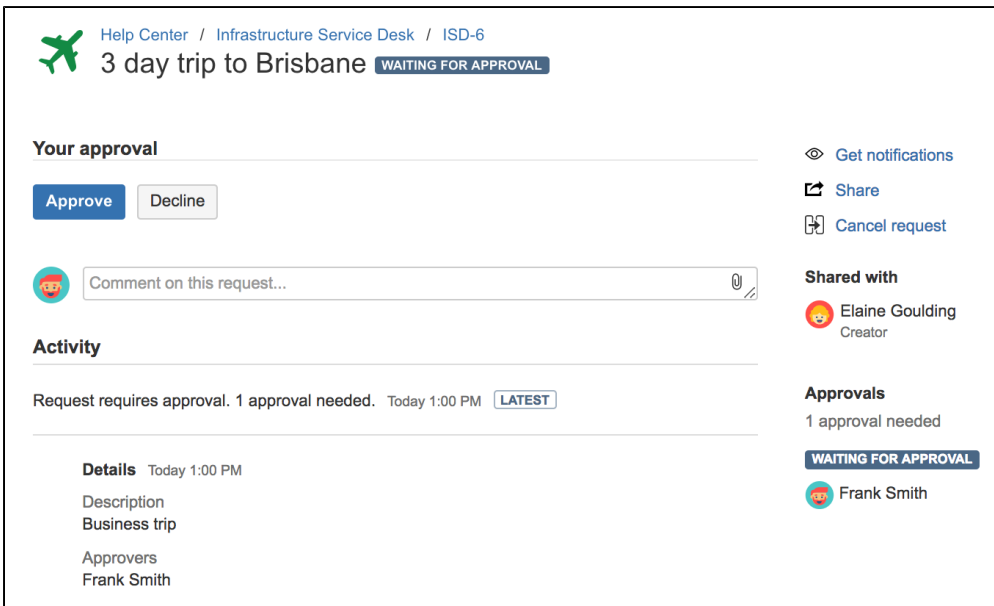
Setting up approvals

Some requests might need approval before your team can work on them. For example, a manager might approve leave requests or an IT manager might approve new system accounts. People don't need a Jira Service Management license to approve requests; they just have to be customers of the service desk project.

 **Pro tip:** Fast-track the time it takes approvers to action pending requests, by inserting **Request details** and **Approval buttons** to your approval notifications template, if they aren't there already. See [Managing service project notifications](#).

On this page:

- [Setting up an approval step](#)
- [Best practice customizations](#)
- [Auto-approve requests](#)
- [Approval FAQs](#)



The screenshot shows a customer portal interface for a request titled "3 day trip to Brisbane" with a "WAITING FOR APPROVAL" status. The interface includes a "Your approval" section with "Approve" and "Decline" buttons, a comment field, and an "Activity" section showing the request requires approval. On the right, there are options for "Get notifications", "Share", and "Cancel request", along with a "Shared with" section listing "Elaine Goulding" as the creator and "Frank Smith" as an approver. A "WAITING FOR APPROVAL" badge is also visible in the bottom right corner.

Approvers view in the customer portal

How it works:

1. A customer creates a request on the customer portal, and selects the approver required by entering the user name or email address. Alternatively, you can hide the approver field from the customer, and set up a defined list of approvers who are required for that request type.
2. When the request enters the approval status the approver will receive an email.
 - a. If the Request details and Approval buttons variables **have been added** to the approval notifications template, they can view the full details of the request and take action from within the email.
 - b. If the Request details and Approval buttons variables **have not been** added to the approval notifications template, they can view and action the request through the service desk customer portal.
3. The approver can Approve or Decline the request and add an optional comment. The customer receives a standard notification when the request is transitioned out of an approval step, and when the approver leaves a comment.
 - a. If declined, the request moves to the next status in the workflow.
 - b. If approved, the request moves to the next status in the workflow, and an agent is able to work on it.

Setting up an approval step

To set up an approval step on a workflow for your project, you need to have the Jira administrator [global permission](#).

Here are the steps to get approvals working for your project:

- Jira Service Management creates the **Approvers** custom field automatically. If you want to use another field, make sure you have a user picker custom field available in your Jira instance and on the screens used by your project - this is used on the approval step.
- Add the same user picker custom field to your request type if you want your customers to choose the approver - this will provide the field for your request that allows an approver to be selected.
- Configure the approvals step on the workflow - this allows you to decide if you require one or multiple approvers, and what happens when the approval is declined or approved.

Detailed steps on [setting up an approval](#) are available in the Jira administration documentation.

If you set up your approval step on a status with only two outgoing transitions, they will be used for Approve and Decline. In this case, agents can view requests that require approval, and can modify the approver if required, but they can't change the status until the approver has actioned the request.

If you set up the approval step on a status that has more than two transitions, an agent will be able to transition the request using any of the other transitions that aren't defined as the Approve or Decline transitions. This means the approval step is not enforced.

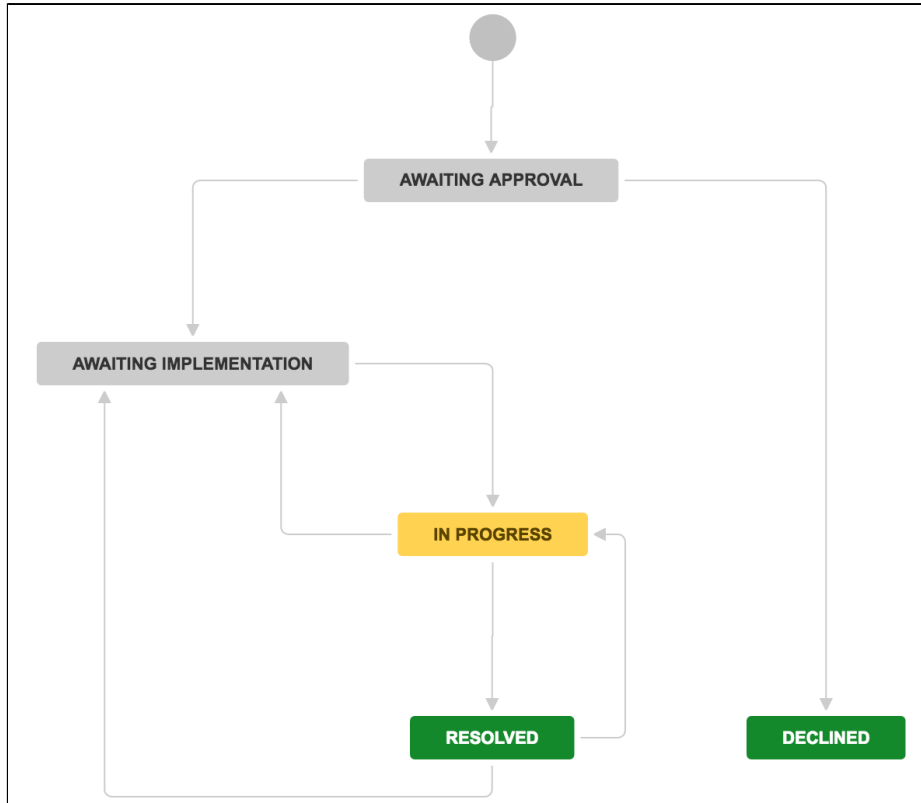
In certain situations, you may even want to add additional approval steps. For example, if a request needs to be approved by your manager first, and then approved by your finance department. Below are some examples of how you might use approvals:

Request type	Approval field	Approver	Notes
Office equipment	Visible on customer portal	Customer selects approver	The customer should select their manager for approval, as each customer could have a different manager.
Computer software	Hidden from customer portal	Set list of approvers	You set the list of approvers, this could be several members of your finance team.
Flights	Two approval steps, one visible, one hidden	First approver is selected by customer. Second approver is a set list	The customer can select the first approver, which could be their manager who approves the business case for their trip. The second approver is from a set list (maybe a finance team) that approves the payment of the flights.

Best practice customizations

You can make the approval process more awesome by making some simple customizations to your service project, and that means both you and your customers have a better experience.

1. Make sure you make the name of your multi-user picker custom field, customer friendly on your request. And add a useful help tip, like how many approvers may be required. This will help ensure your customers provide all the correct information first time. Read up on [customizing the fields of your requests](#) for more information on how to do this.
2. When the approval of a request is declined, consider closing the request straight away. This means that if you're setting up a custom approval step, you should ensure that the transition you select for decline leads to a status in the Done category, and the transition has a post function to set the resolution status.



Workflow showing Declined transition leading to Declined status in the Done category

- It's good practice for your approvers to add a comment telling the customer why their request has been declined, and what their next steps should be; for example, the customer should open a new request and provide more information regarding their requirements.

Auto-approve requests

You can use automation rules to auto-approve requests that meet certain conditions. For example, you might auto-approve hardware requests that are under a certain amount.

To set up an auto-approval, create an automation rule with the THEN action auto-approve. For example, any hardware requests created for an item that costs less than \$20 are automatically approved, or any software requests over \$10,000 are automatically declined.

In the example below, we've set up a rule for our customer reimbursement requests, and if the Customer value is under \$50, the request is automatically approved.



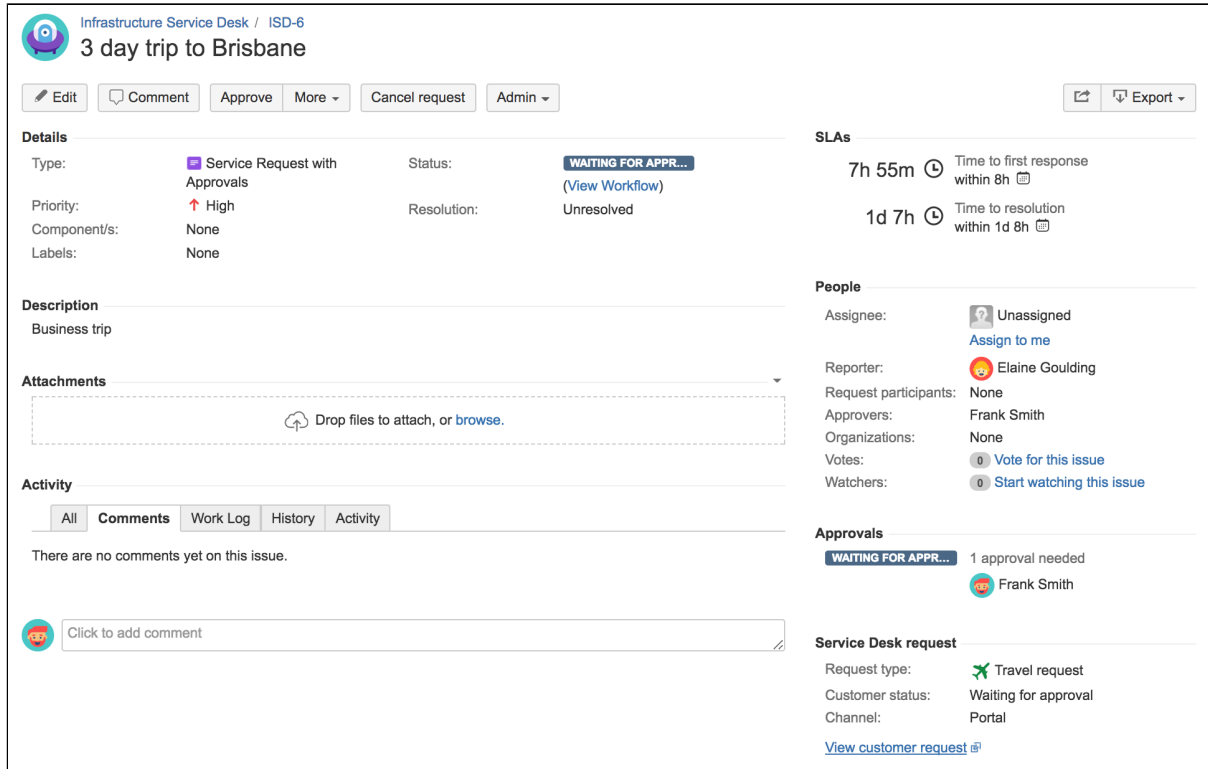
For more information, see [Automating your service project](#).

Approval FAQs

Make sure you've added the user picker custom field that's used in the approval step on the workflow to your [request type](#), and it's visible to customers. If it's still not showing, or it's not listed as a field you can add to your requests, check with your Jira administrator to ensure the field is still available on your project's issue screens.

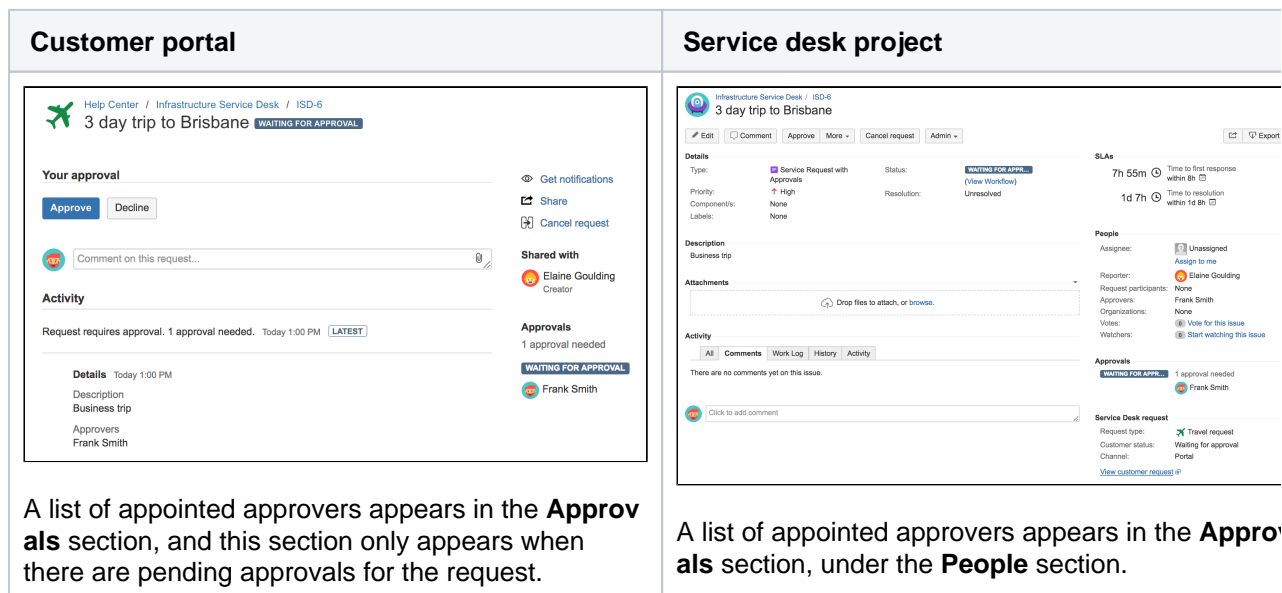
A request will only show as closed when the resolution has been set, so the transition you use to decline the request needs to also set the resolution field. You can achieve this by adding a post function to the transition in your workflow that sets the resolution field of your request. You read up more on post functions on the administration [advanced workflow](#) page.

If you need to change the approvers on a request for any reason, you need to edit the user picker custom field that was added in the approval step. View the request in your service project, and the editable user picker custom field is displayed in the **People** section of the request. You can make any changes you need to make inline. Note that there is also an **Approvals** section that lists all approvers; however, you can't make any edits here.



If the field isn't showing, you may need to get a Jira administrator to check the field is still available on your project screens.

In both your customer portal and your service project, you'll see a list of approvers in the **Approvals** section.

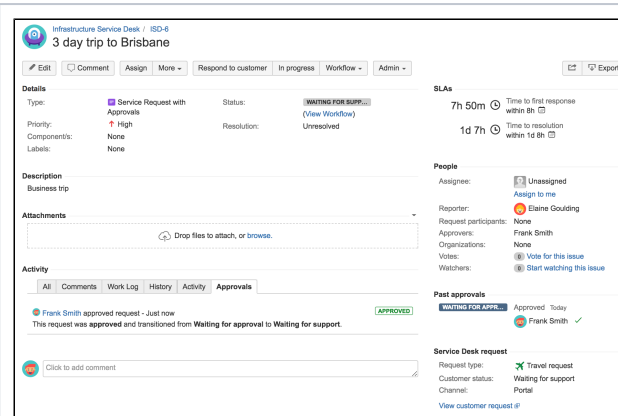


A list of appointed approvers appears in the **Approvals** section, and this section only appears when there are pending approvals for the request.

A list of appointed approvers appears in the **Approvals** section, under the **People** section.



When a request is approved, the **Approvals** section disappears, and details about the approval will be added to the **Activity** section, e.g Your request was **APPROVED** and the status changed to **Waiting for support**.



When a request is approved...

- The approved request displays in the **Past approvals** section, with ✓ beside the approver's name.
- If multiple approvals are required, the request is approved when the minimum number of approvals is reached.
- The **Approvals** tab appears in the **Activity** section, and shows the all the activity involving approvals for the request.

Make sure you add [descriptive titles and help text](#) for the approvals field on your portal. This should include details on who the customer should add (such as their manager, or a member of their IT team), and how many approvers. Depending on your [customer access settings](#), your customers may be able to select users or other customers from a drop-down list. Take this into account when adding your title and help text.

Setting up service project reports

Reports allow teams to look at trends in their project such as the amount and types of requests received, and how they're resolved. We recommend that all teams use reports, especially those who use SLAs. [Read more about reporting on SLAs.](#)

To view or create reports, go to **Reports** from your service desk project's sidebar. *You must be a project administrator to create or edit reports.*

i If the JQL filters in your reports use priorities, make sure these filters include all the priorities defined in the associated priority scheme. See [Associating priorities with projects](#) for more details.

On this page:

- [Compare requests created vs requests resolved](#)
- [Default Jira Service Management reports](#)
- [Create custom reports](#)
 - [Customer satisfaction report](#)
 - [Requests created per channel report](#)
 - [Average resolution time by issue type report](#)
 - [Regional trends report](#)

Compare requests created vs requests resolved

The most common way to measure your service team's health is by comparing how many requests come in with how many your team can resolve.

Resolution trends like this can answer questions like:

- Was this week's volume of requests a one-time occurrence or the start of a trend?
- Do we support a service that causes more issues than it's worth?
- Is our service project scaling with our business or do we need to add staff?

We include this report by default because it's the quickest way to check the health of your service project. You can find the other default reports in the **Reports** sidebar.

To compare requests created vs resolved:

1. From your service project, go to **Reports**.
2. Select the **Created vs Resolved** report in the sidebar.
3. In the dropdown field below the graph, choose the date range you're interested in.
4. Click on a number count in the table to drill down into individual requests.



You can also create a CSV report of the dataset by choosing **Export CSV** in the top right-hand corner.

Default Jira Service Management reports

Here are the reports we include by default:

Report name	Details
Workload	The number of requests assigned to your agents
SLA goals	How your team is tracking towards each of the SLA goals you have set
Satisfaction	The average customer satisfaction rating for your team
Article usage	<p>The number of times your customers viewed knowledge base articles in the portal and found them helpful</p> <p>✔ Learn which articles solved requests and how many views each got (as well as other useful metadata) by clicking on the number count in the table below the graph.</p>
Article effectiveness	<p>The number of requests resolved with and without knowledge base articles</p> <p>✔ Learn which articles were shared by agents and which requests were resolved without an article (as well as other useful metadata) by clicking on the number count in the table below the graph.</p>
Created vs resolved	Compares the number of requests created and resolved over time

Time to resolution	Compares the length of time taken to resolve requests of type or priority
SLA met v breached	Compares the number of requests that have met or breached an SLA goal
Resolution by component	Compares the resolution times for each component (for basic service desks only)
Incidents reports by priority	Compares the priority of incidents your customers have reported (for IT service desks only)

Besides the above, you can dig deeper. *Why* does your performance look like this? Create custom reports for your service desk and explore this question.

Create custom reports

Custom reports, where you define the series, help you to find out more about why your performance is the way it is.

Series are a set of data points used to make reports. For example, a series could be the number of requests received on day 1, 2, 3 and so on for the past week. This would compare the difference in the number of requests received on each day of the week. While they can point out trends on their own, they're more powerful when plotted together.

To create a custom report:

1. From your service project, go to **Reports**.
2. Select **New report**.
3. Choose a report name that you and your team will understand. For example, *High priority issues* would be a report that shows data on prioritized issues.
4. Click **Add a series**.
5. Fill in the following details: **Series**, **Label**, **Color**, and, optionally a filter.
6. Click **Add**.
7. Add more series to compare values.
8. Click **Create**.

The screenshot shows a configuration window for a custom report. It includes the following elements:

- Series:** A dropdown menu with "Average rating" selected.
- Label:** A text input field containing "Bugs".
- Color:** A grid of 24 color swatches. The first swatch (red) is highlighted with a blue border.
- Filter by:** A text input field containing "issuetype = 'Bug'".
- Basic:** A button located at the bottom right of the configuration area.

Check out the recommended reports below to explore how reports benefit your organization. See examples of what series and their associated JQL filters may be useful to you. Read more about [Advanced searching through JQL](#).

Customer satisfaction report

One of the best measures of performance is your customers' happiness. Jira Service Management comes with reports on customer satisfaction, but a custom report can help you find more trends. First, be sure to collect customer satisfaction information on your requests. [Learn how to enable customer satisfaction feedback](#).

Use the average rating customer satisfaction series to see how your team performs. For example, you can use issue types to investigate sections of your business.

To create a report to see if your customers are satisfied:

1. From your service project, go to **Reports**.
2. Click **New report**.
3. Enter "Customer satisfaction" as the **Name**.
4. Add the following series, then click **Create**:

- **Series**= Average rating
- **Label**=Bugs
- **Filter by**(advanced) = issuetype= "Bug"

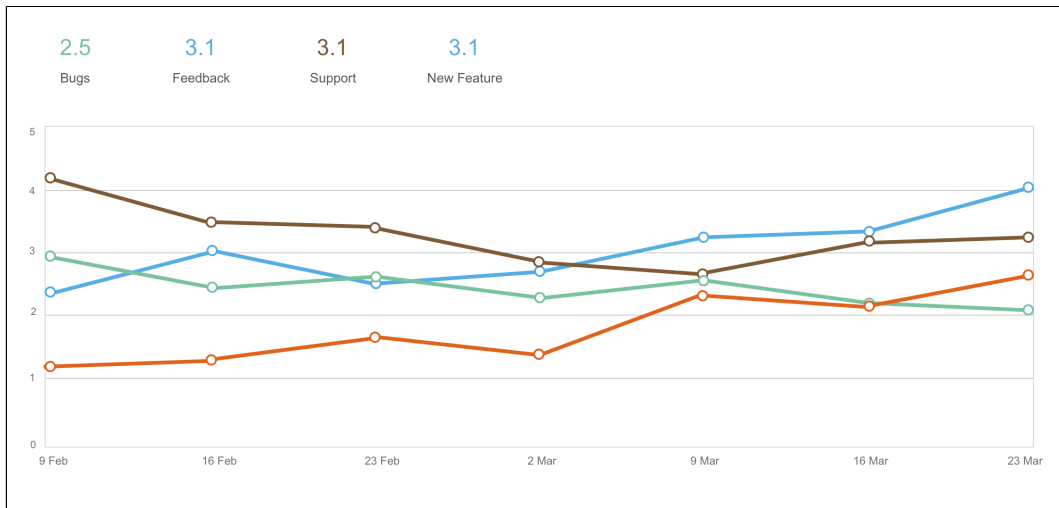
- **Series**= Average rating
- **Label**=Feedback
- **Filter by**(advanced) = issuetype = "Feedback"

- **Series**= Average rating
- **Label**= Support
- **Filter by**(advanced)= issuetype = "Support"

- **Series**= Average rating
- **Label**=New feature
- **Filter by**(advanced) = issuetype = "New feature"

You might find some interesting results. For example, your response to feature requests may please your customers, but, they aren't happy when they raise requests about billing. If you select a data point in the report you can look for feature words like "payment" or "credit card".

Details like these can expose your customers' pain points. Maybe your organization needs an easier to use billing form. Or perhaps you can be clearer about how much your products or services cost, or which credit cards you accept.



Requests created per channel report

Monitoring how your customers submit requests can give you useful insights. Are you getting more requests from email than from your customer portal? You can create a report to see how many requests customers create in each channel.

To create a report to track requests sent by portal and email:

1. From your service project, go to **Reports**.
2. Click **New report**.
3. Enter "Requests created per channel" as the **Name**.
4. Add the following series and click **Create**:

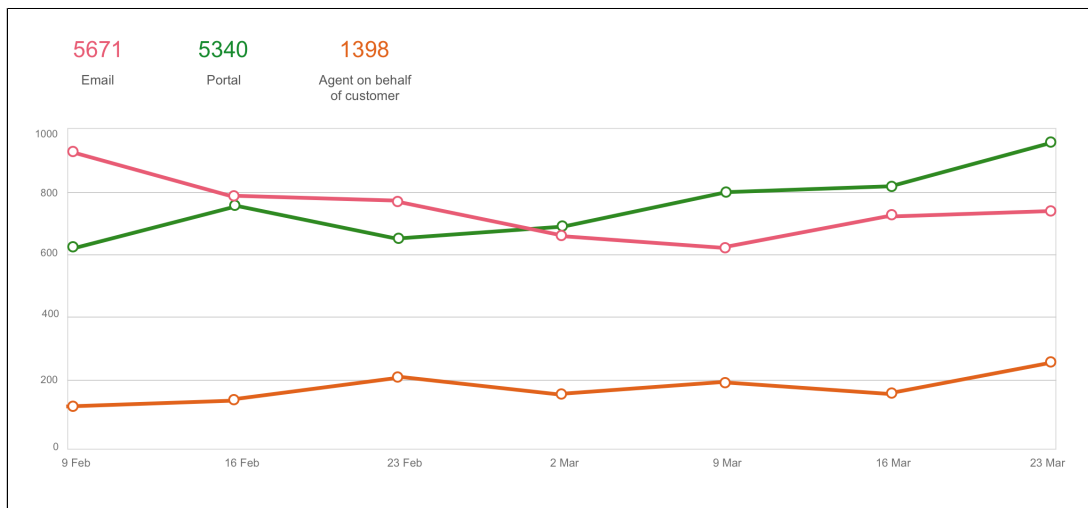
- **Series**= Created
- **Label** =Email
- **Filter by**(advanced)= request-channel-type = email

- **Series**= Created
- **Label** =Portal
- **Filter by**(advanced)= request-channel-type = portal

- **Series**= Created
- **Label** =Agent on behalf of customer
- **Filter by**(advanced)= request-channel-type = Jira

The last series catches issues agents raise outside the portal.

How your customers request help might surprise you. You might find that your agents raise more and more requests on your customers' behalf. If so, you can find ways to direct customers to your portal or email channels. That way your agents have more time to resolve issues, rather than raise them.



Average resolution time by issue type report

Jira Service Management tracks requests by time. The time it takes your team to resolve a type of issue can show trends in your teams' efficiency.

To create a report to see your average resolution by issue type:

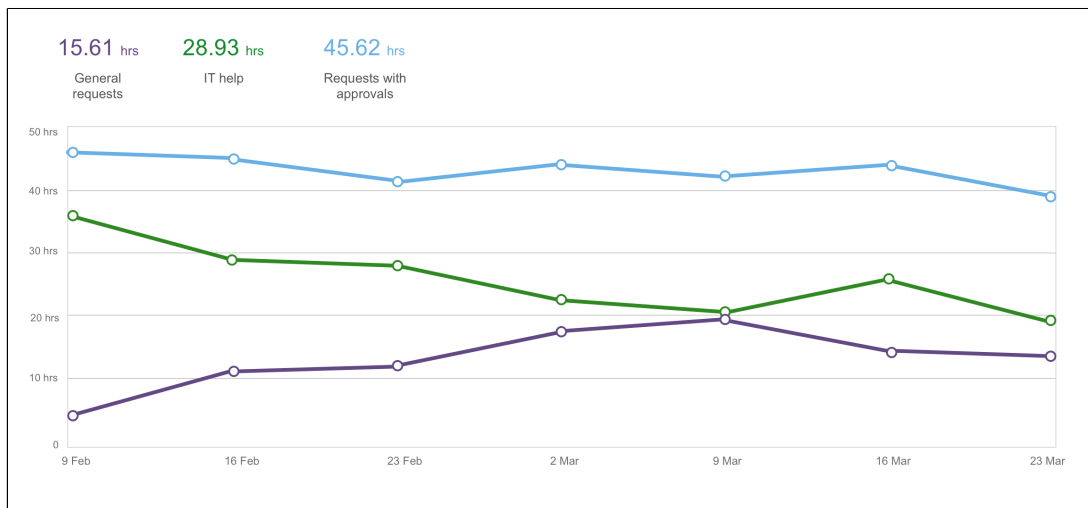
1. From your service project, go to **Reports**.
2. Click **New report**.
3. Enter "Resolution by issue type" as the **Name**.
4. Add the following series and click **Create**:

- **Series**= Time to resolution (Avg.)
- **Label**= General requests
- **Filter by**(advanced) = "Customer Request Type" = "General requests"

- **Series**= Time to resolution (Avg.)
- **Label**= IT help
- **Filter by**(advanced) ="Customer Request Type" = "IT help"

- **Series**= Time to resolution (Avg.)
- **Label**= Requests with approvals
- **Filter by**(advanced) ="Customer Request Type" = "Request with approval"

You may find that IT help requests take more of your teams' time than general requests. Take into account how many IT help requests come through your service project. With this info, you can better divide your agents, and their time, to make your customers happier.



Regional trends report

If you service more than one location, you can cut out noise by viewing your regional performance. Start by adding labels to requests, identifying and helping to sort them into regions.

For example, if your business operated in New York and Rio de Janeiro, your service project agents can add a location label to requests from each region.

To create a report to see how many requests come from each location:

1. From your service desk project, go to **Reports**.
2. Click **New report**.
3. Enter "Requests by region" as the **Name**.
4. Add the following series and click **Create**:

- **Series**= Created
- **Label** =New York
- **Filter by**(advanced)= labels = ny

- **Series**= Created
- **Label** =Rio
- **Filter by**(advanced)= labels = rio

If you see an increasing trend in one location, you may decide to shuffle around resources. Maybe you've opened a new location without a dedicated service project team member, and the new location finds it difficult to ramp up operations. Perhaps you need to send someone to provide training? Or, maybe there's a language barrier with your knowledge base. You may consider providing support articles in more than one language.

You might see the opposite, a decline in requests coming from one location. Are people abandoning your service project? Do you need to make it clear that the service project operates for all locations?

Default service project configuration

Use this page as a reference for the default configuration of your service projects, including custom fields, permissions, and database tables.

- [Custom fields](#)
- [Request types, issue types, and workflows](#)
- [Project permissions](#)
- [Security types](#)
- [Database tables](#)

Custom fields

If required, Jira Service Management will create the following custom fields:

Custom field	Type	Notes
Viewport Origin	String value, storing the 'Portal' and 'Request Type' if a request was created through the customer portal.	Issues must have this field to be a service desk request.
Time to resolution	An SLA field, stored in JSON format.	This field stores SLA information for time until a request's resolution is set. See Setting up SLAs for more information.
Customer Request Type	String value	Issues must have this field to be a service desk request.

Request types, issue types, and workflows

The default issue types, request types, and workflows are different for each service desk project type. When you create a new service project, you can view these defaults by selecting Request types, Issues types, or Workflows from the **Project settings** menu.

Project permissions

At installation time, Jira Service Management creates a project permission called **Jira Service Management agent access**. Users who require full access to service desk projects or functionality need to have this permission.

This page shows the permission configuration for a standard service project permission scheme.

- To see an overview of how permissions are set up for a service project, see [Permissions overview](#).
- If you want to customize the permission scheme, see [Customizing Jira Service Management permissions](#).
- If you run into permission-related problems, see [Resolving Jira Service Management permission errors](#).

Security types

Jira Service Management introduces the **Service Desk Customer - Portal Access** security type. A security type is a concept that allows restriction of users to certain permissions, examples of security types include [project roles](#) and [groups](#).

Service Desk Customer - Portal Access is a special security type that only applies to users while they are viewing the customer portal; it was created specifically to allow customers to use the customer portal without giving them access to the internal service project view and your other Jira applications.

Database tables

When you set up Jira Service Management, the following tables will be created in your Jira application database.

General Jira Service Management:

- AO_54307E_AGENTSIGNAUTRES
- AO_54307E_ASYNCUPGRADERECORD
- AO_54307E_CAPABILITY
- AO_54307E_CONFLUENCEKB
- AO_54307E_CONFLUENCEKBENABLED
- AO_54307E_CONFLUENCEKBLABELS
- AO_54307E_CUSTOMGLOBALTHEME
- AO_54307E_CUSTOMTHEME
- AO_54307E_EMAILCHANNELSETTING
- AO_54307E_EMAILSETTINGS
- AO_54307E_GOAL
- AO_54307E_GROUP
- AO_54307E_GROUPTOREQUESTTYPE
- AO_54307E_IMAGES
- AO_54307E_METRICCONDITION
- AO_54307E_PARTICIPANTSETTINGS
- AO_54307E_QUEUE
- AO_54307E_QUEUECOLUMN
- AO_54307E_REPORT
- AO_54307E_SERIES
- AO_54307E_SERVICEDESK
- AO_54307E_STATUSMAPPING
- AO_54307E_THRESHOLD
- AO_54307E_TIMEMETRIC
- AO_54307E_VIEWPORT
- AO_54307E_VIEWPORTFIELD
- AO_54307E_VIEWPORTFIELDVALUE
- AO_54307E_VIEWPORTFORM

Jira Email Processor Plugin:

- AO_2C4E5C_MAILCHANNEL
- AO_2C4E5C_MAILCONNECTION
- AO_2C4E5C_MAILGLOBALHANDLER
- AO_2C4E5C_MAILHANDLER
- AO_2C4E5C_MAILITEM
- AO_2C4E5C_MAILITEMAUDIT
- AO_2C4E5C_MAILITEMCHUNK
- AO_2C4E5C_MAILRUNAUDIT

Automation:

- AO_9B2E3B_EXEC_RULE_MSG_ITEM
- AO_9B2E3B_IF_CONDITION_CONFIG
- AO_9B2E3B_IF_COND_CONF_DATA
- AO_9B2E3B_IF_COND_EXECUTION
- AO_9B2E3B_IF_EXECUTION
- AO_9B2E3B_IF_THEN
- AO_9B2E3B_IF_THEN_EXECUTION
- AO_9B2E3B_PROJECT_USER_CONTEXT
- AO_9B2E3B_RSETREV_PROJ_CONTEXT
- AO_9B2E3B_RSETREV_USER_CONTEXT
- AO_9B2E3B_RULE
- AO_9B2E3B_RULESET
- AO_9B2E3B_RULESET_REVISION
- AO_9B2E3B_RULE_EXECUTION
- AO_9B2E3B_THEN_ACTION_CONFIG

- AO_9B2E3B_THEN_ACT_CONF_DATA
- AO_9B2E3B_THEN_ACT_EXECUTION
- AO_9B2E3B_THEN_EXECUTION
- AO_9B2E3B_WHEN_HANDLER_CONFIG
- AO_9B2E3B_WHEN_HAND_CONF_DATA

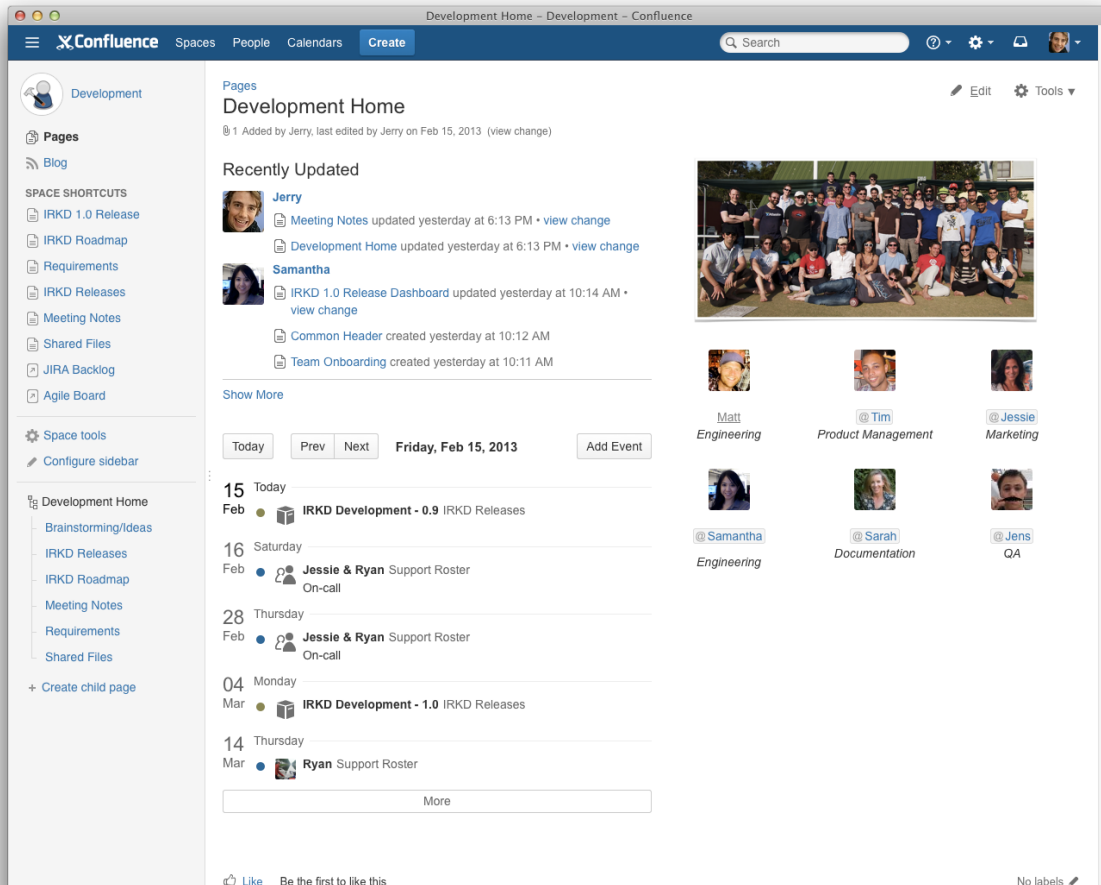
Jira Timed Promises Plugin:

- AO_F1B27B_HISTORY_RECORD
- AO_F1B27B_KEY_COMPONENT
- AO_F1B27B_KEY_COMP_HISTORY
- AO_F1B27B_PROMISE
- AO_F1B27B_PROMISE_HISTORY

Using Jira applications with Confluence

What is Confluence?

Confluence is a content creation and collaboration platform that connects teams with the content, knowledge, and coworkers they need to get work done, faster. Confluence spaces are great for creating and organizing rich content related to Jira projects using Confluence pages: meeting notes, project plans, requirements documents, release notes, roadmaps, and more.



Why use Confluence with Jira?

Here are some of the reasons we think you might like to add Confluence to your Jira instance:

For a...	You can...
Bug	Create a knowledge base article to document a workaround for a bug.
New Feature	Create a product requirements document for a new feature.
General Jira Use Case	Document and collaborate with your team on an issue in Confluence.

And here are just a few of the things Confluence allows you to do:

- Collaborative commenting, especially through the use of @mentions
- Share pages
- Watch pages
- Form a 'team' network and let them know what you are doing via a status update
- Add images, picture galleries, videos, and more
- Enable various content macros

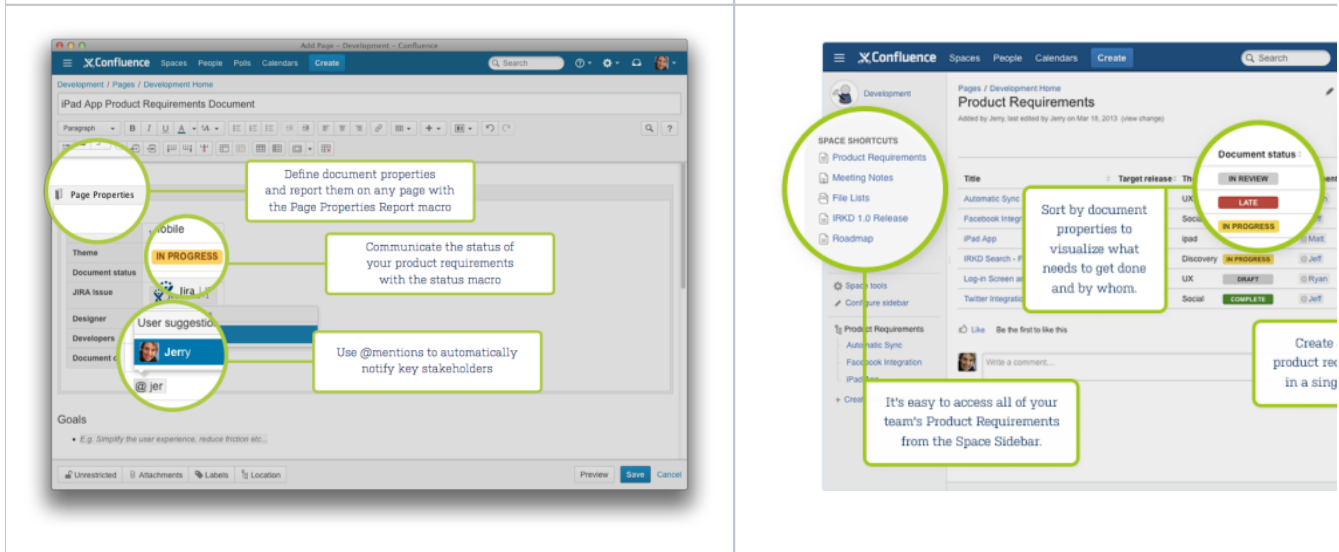
Confluence features for Jira users

Here are some of the best features of Confluence that Jira users would benefit from.

Define product requirements

Many of our customers write product requirements using Confluence to plan new product features. The Requirements Blueprint helps development teams create, discuss, and organize their product requirements to link your product requirements in Confluence to

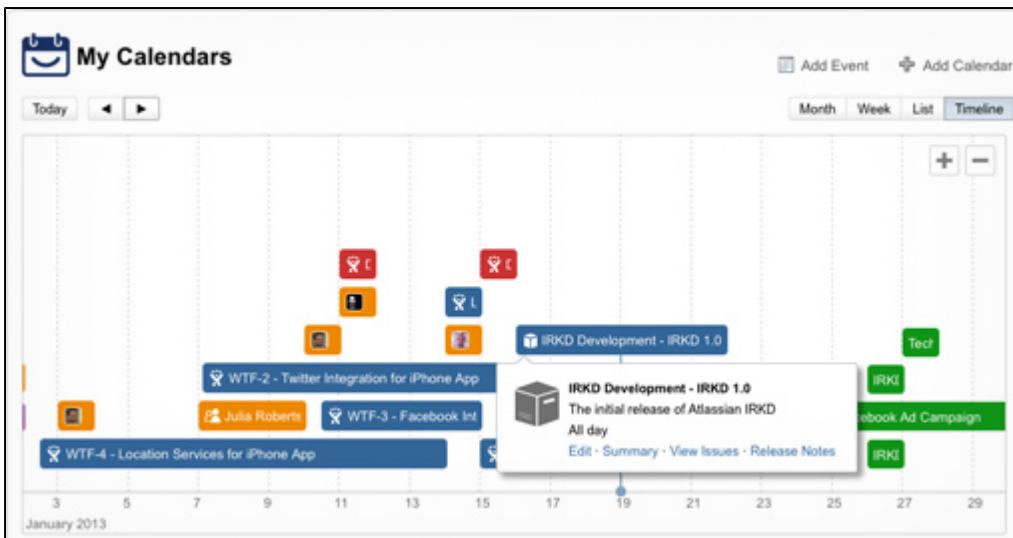
For more information, see [Blueprints](#).



Team Calendars: Your Birds-Eye View of Jira

Surface everything your development team is working on in Jira to the teams that live in Confluence with Team Calendars.

- **Timeline Calendar:**View plans 3 months ahead of time.
- **JQL Support:**Track your versions, issues, and agile sprints.
- **Date Ranges:**Visualize issues over time to understand upcoming workload.

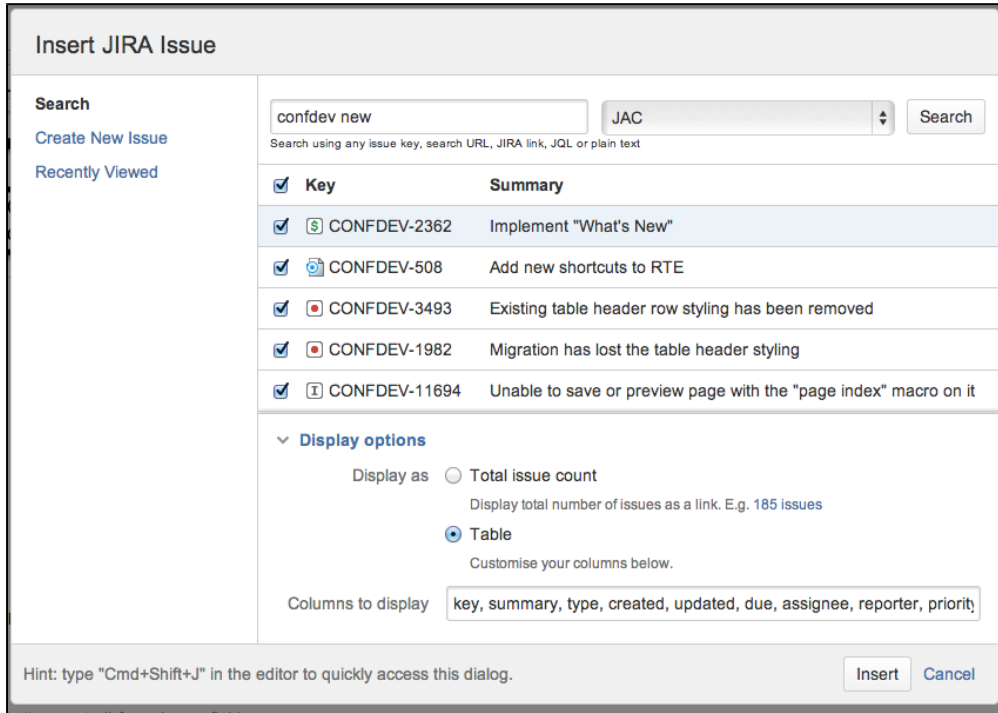


To install this feature, please visit [Atlassian Marketplace](#).

Insert issues on any Confluence page using the Jira Issues macro

Any Jira search result can be embedded in a Confluence page using the [Jira Issues macro](#) with your choice of included fields and field ordering. With the Jira Issues macro, you can:

- Display a table of issues on your page, based on the results of a search using [Jira Query Language \(JQL\) syntax](#) or a Jira URL.
- Display a single issue from the Jira site, or a subset of selected issues from your Jira search results.
- Display a count of issues from the Jira site.
- Create a new issue on the Jira site and display that issue on your page.



Autoconvert pasted issue links

Autoconvert makes producing reports of issues, backlogs, and tasks as easy as copy and paste. With Jira and Confluence connected, you can paste individual issues or Jira query URLs into the editor and watch them immediately transform into the Jira Issues macro.

Automatic links

Whenever an issue is mentioned in a Confluence page using the Jira Issues macro, Jira will create an issue link to that page for you, automatically. Specs to issues, knowledge base articles to support tickets, project outlines to tasks it all works.

Gadgets

You can embed a Confluence activity stream or a Confluence page in Jira's dashboard. Likewise, Jira gadgets can be rendered on a Confluence page.

Working on service projects

If you are an agent working on a Jira Service Management project, you're in the right place!

If this is the first time you have used Jira Service Management, check out [Getting started for service agents](#) for a brief introduction to your new workspace.

If you're familiar with Jira Service Management, use the search bar below to find any needed information.

Search the topics in 'Working on service projects':

Working on issues

- [Work with issues](#)
- [Edit and collaborate on issues](#)
- [Attach files and screenshots to issues](#)

Tracking your work

- [Keep on top of SLAs](#)
- [Set up dashboards](#)

Serving your customers

- [Raise requests on behalf of customers](#)

Working with issues

In Jira Service Management, customer requests are automatically triaged into queues, so you can easily find the issues you need to work on. If you are ready to jump in and learn more about working on and managing customer issues, you're in the right place. This page introduces you to the concept of an issue. You can then learn more about creating, editing, and collaborating issues in the Next steps section.

On this page:

- [What is an issue?](#)
- [Next steps](#)

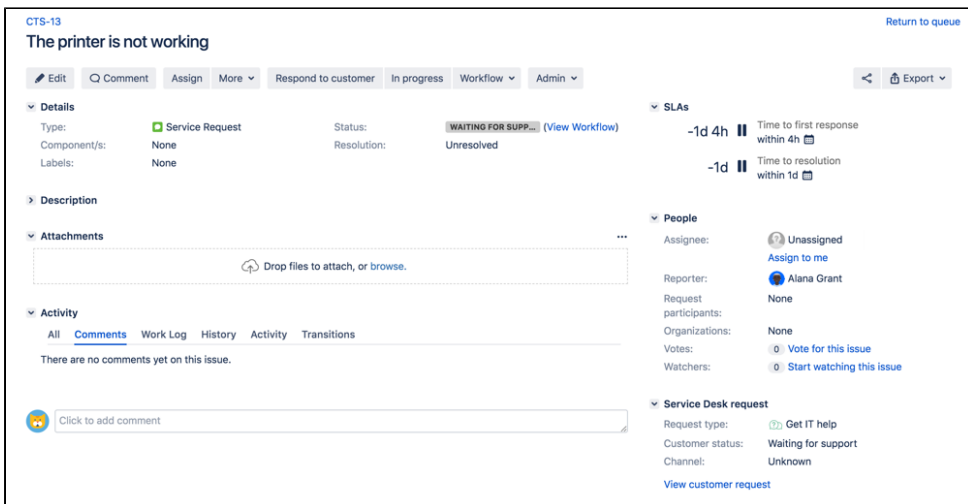
What is an issue?

Different organizations use Jira applications to track different kinds of issues, which can represent anything from a software bug, a project task, to a leave request form.

In Jira Service Management, an issue is a packet of work that agents work on. In an IT service desk, it represents an incident, a change, and a service request, etc. For example, a customer request of "Our printer is not working" appears as follows in the customer portal:



As an agent, you will pick the issue up internally in the service desk project to work on and it will look like the following:



The **History** tab of an issue records the following information: creator of the issue (this may be the same as the reporter, but can be distinct), changes to an issue field, attachment of a file, deletion of a comment, deletion of a worklog, creation or deletion of an issue link.

The **Activity** tab has the same information, plus additional information, such as comments. However, this may load more slowly, especially if there has been a lot of activity on the issue.

Next steps

Check out the following pages to reach issue ninja status:

- [Creating issues and sub-tasks](#)
- [Attaching files and screenshots to issues](#)
- [Editing and collaborating on issues](#)
- [Logging work on issues](#)

Attaching files and screenshots to issues

To share information with your customers, you can attach documents, images, and screenshots to your Jira Service Management issues. You can also restrict attachments to be viewed by your internal team only.

On this page:

- [Before you begin](#)
- [Add attachments](#)
- [Sort and manage attachments](#)
- [Access ZIP file contents](#)
- [Capture and attach screenshots](#)

Before you begin

A Jira admin must enable specific user permissions so that you can add attachments and screenshots to issues. The most common permissions are described below. To learn more about this check out [Configuring file attachments](#).

Jira admin set permissions

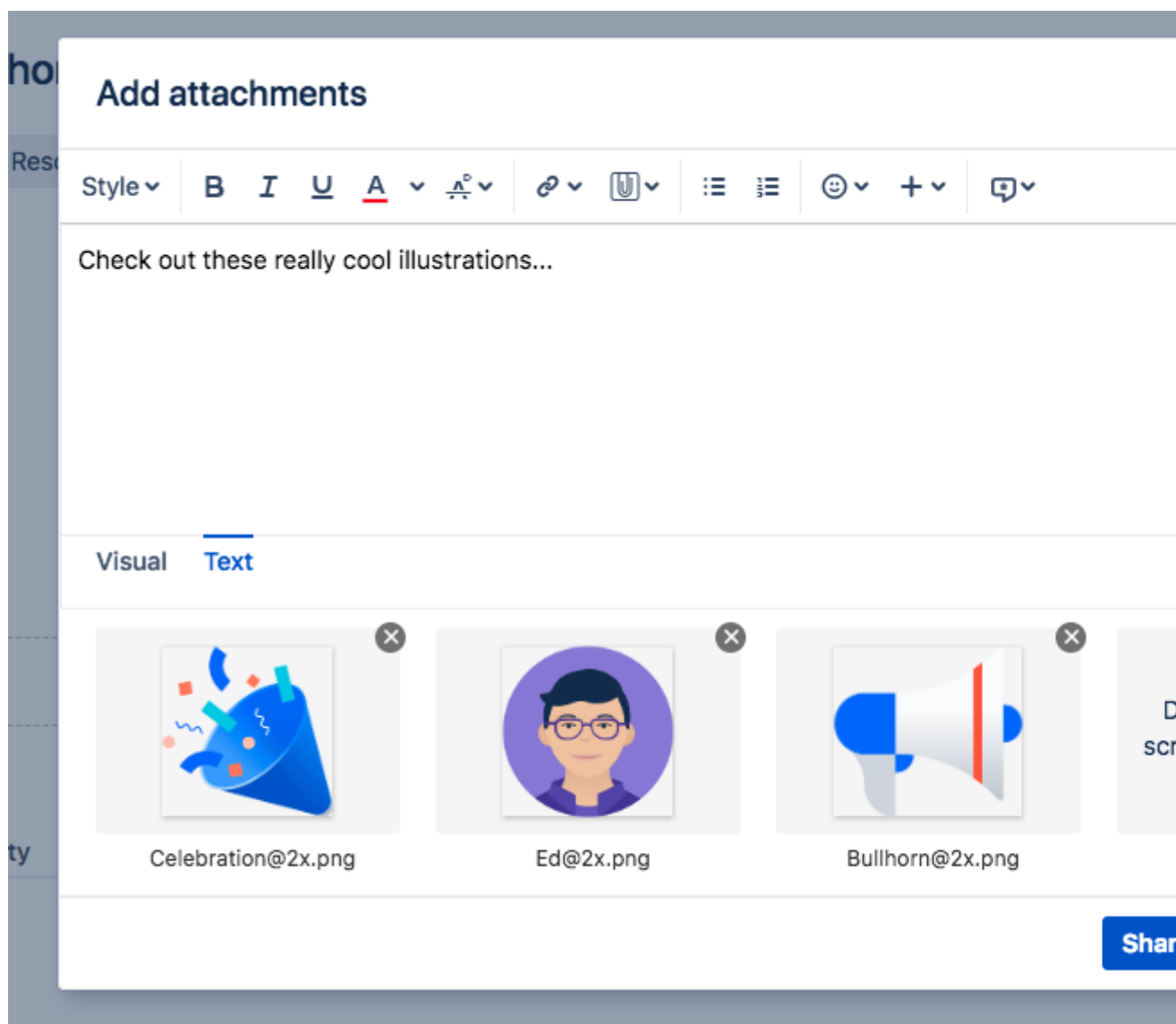
- You can attach files and screenshots if your Jira admin has enabled file attachments.
- You need the **Create attachments** permission in the appropriate projects.
- If your Jira admin has disabled thumbnails in Jira's attachment settings, the image files will appear as a list.
- If your Jira admin has disabled ZIP support in Jira's attachment settings, the attachments feature will not be available. You must download the file to your computer before accessing its individual files.
- To remove attachments from an issue, you need one of the following project permissions in that issue's project:
 - **Delete own attachments** to delete files that you have added to the issue.
 - **Delete all attachments** to delete files that anyone has added to the issue.

- If you're using Google Chrome, Mozilla Firefox, or Internet Explorer 11, attaching screenshots relies on HTML5 compatibility. Safari is not supported.

Add attachments

You can add files and images to any issue in your service desk project.

1. When working on an issue, drop a file onto the issue or select **browse**.
2. Add a comment with more information about the attachment (optional).
3. Share the file and comment with your customer, or with your internal team only.



When adding or editing a comment you can also select to add attachments. In this case, you'll see wiki markup added to the comment field. As soon as you share your comment, you'll see the file preview.

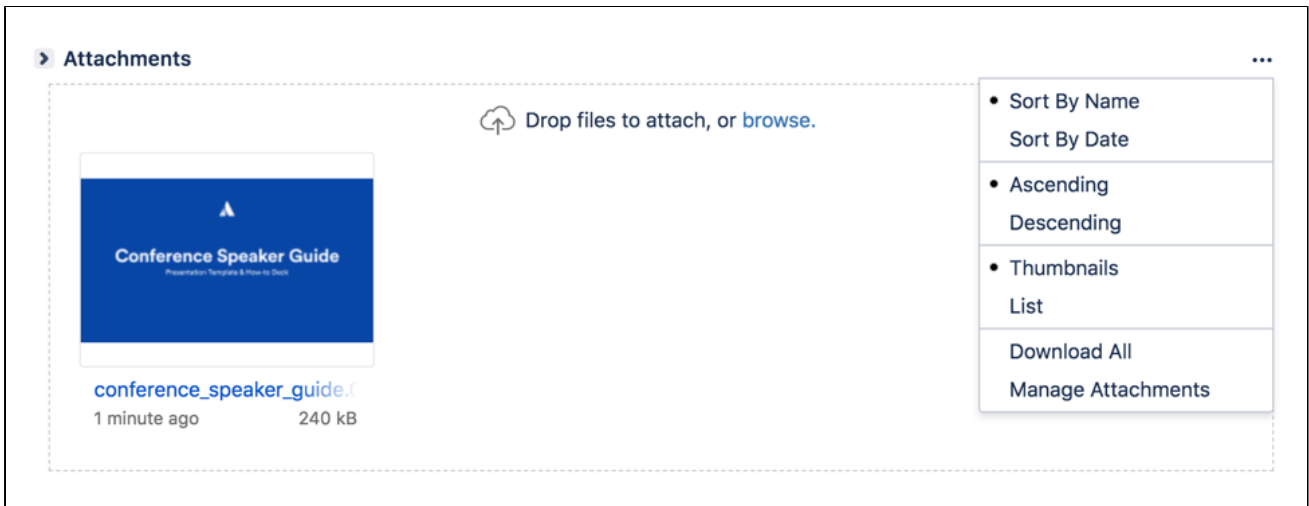
Some considerations for attached files:

- File formats: GIFs, JPGs, PNGs
- File names can't contain any of these characters: '\', '/', '\\', '%', ':', '\$', '?', '*'.
- By default, the maximum size of any one file is 10MB, although this limit can be customized by your Jira admin.

Sort and manage attachments

The attachments section of the issue displays a list of options to sort, manage, and download attachments.

1. Select the three dots to the right of the attachments section to open the menu.
2. Reorder the attachments according to a selected criteria. This criteria will be applied to all issues in your project, and will be lost once you log out.
3. To remove attachments from the issue, select **Manage attachments** or hover over the attachment and select the bin icon.



Access ZIP file contents

You can view the contents of a ZIP file (including '.zip' or '.jar' file name extensions) in the attachments section.

1. Click the three dots and select **List**.
2. In list view, click the arrow icon in front of the zipped file's name to view and download its individual files.
3. To download the entire zip file, click **Download Zip**.



Capture and attach screenshots

You can capture a screenshot to the system clipboard and paste it directly onto an issue.

1. Capture a screenshot using your system keyboard shortcut.
2. Paste the image from your clipboard onto the issue using your system keyboard shortcut or right-click menu. The **Attach screenshot** dialog will display.
3. Enter a unique filename for each file.
4. Select **Upload**.

Creating issues and sub-tasks

The building blocks of any project are issues. Issues act as the packets of work that travel through their respective workflows within their projects, until the work is completed. An issue may also have sub-tasks that can be assigned and tracked individually, as well as issue level security to restrict the issue to select members of your team.

On this page, you'll learn more about creating and converting issues and sub-tasks, and setting issue level security. If you are looking to import multiple issues (and sub-tasks) using a CSV file, you can find the import process explained in more detail [here](#).

Before you begin

You need the **Create Issue** project permission for the issue's relevant project.

On this page:

- [Before you begin](#)
- [Creating an issue](#)
- [Cloning an issue](#)
- [Creating a sub-task](#)
- [Converting a sub-task to an issue](#)
- [Converting an issue to a sub-task](#)
- [Restricting access to an issue](#)

Creating an issue

1. Select **Create** at the top of the screen to open the **Create Issue** dialog box.
2. Select the relevant **Project** and **Issue Type** in the **Create Issue** dialog box.
3. Type a **Summary** for the issue and complete any appropriate fields - at least the required ones that are marked by an asterisk.
If you want to access fields that are not shown in this dialog box, or you want to hide existing fields:
 - a. Select the **Configure Fields** button at the top right of the screen.
 - b. Select **Custom** and select the fields you want to show or hide by selecting or clearing the relevant check boxes respectively, or select **All** to show all fields.
When you next create an issue, these selected fields will be displayed.
4. Optional: To create a series of similar issues with the same **Project** and **Issue Type** select the **Create another** checkbox at the bottom of the dialog. Depending on your configuration and the values you may have specified when creating previous issues, some of the fields in the new Create Issue dialog box may be pre-populated. Make sure you check they're all correct before creating the next issue.
5. When you are satisfied with the content of your issue, select the **Create** button.

Cloning an issue


Cloning an issue lets you quickly create a duplicate of an issue within the same project. The cloned issue contains most of the same details stored in the original issue - e.g. Summary, Affects Versions, Components, etc. Other details are not cloned - e.g. Work Log, Comments, Issue history, and Links to Confluence pages. The issue status also returns to the first step of the corresponding workflow, and the resolutions are cleared. The cloned issue can be linked to the original issue, but does not have to be.

Note that, if you don't have the Modify reporter permission, the clone issue will be created with Reporter as the current user cloning the issue.

1. Open the issue you wish to clone.
2. Select **More > Clone**. The **Clone issue** screen will appear.
3. You can edit the clone issue's **Summary** if you want.
4. If applicable to the issue you are cloning, you can also select from these options:
 - **Clone sub-tasks** to copy existing sub-tasks
 - **Clone attachments** to add any existing attachments
 - **Clone links** to add any existing linked issues
 - **Clone sprint values** to copy across the issue's current and closed sprint values
5. Select **Create**.

Creating a sub-task

A sub-task can be created for an issue to either split the issue into smaller chunks, or to allow various aspects of an issue to be assigned to different people. If you find a sub-task is holding up the resolution of an issue, you can convert the sub-task to an issue, to allow it to be worked on independently. If you find an issue is really just a sub-task of a bigger issue, you can also convert an issue to a sub-task.

 You can only create sub-tasks if your administrator has enabled sub-tasks, and has added the sub-task issue type to the project's issue type scheme.

1. Navigate to the issue you would like to be the parent issue of the sub-task you are about to create.
2. Select **More > Create sub-task**. You will see the **Create sub-task** screen.
3. Fill in the details as needed, and then select **Create** at the bottom of the page.

Note that when you create a sub-task, the following values are inherited from the parent task:

- project
- issue security level
- sprint value, if any (only for Jira Software issues)

Tip: You can customize the **Create sub-task** screen to show fields you use most often. To do this, select **Configure fields** at the top right corner of the dialog, and use the **All** and **Custom** links to switch between the default screen and your custom settings. Your changes are saved for future use.

Converting a sub-task to an issue

1. Navigate to the sub-task issue you would like convert.
2. Select **More > Convert to issue**.
3. In the **Step 1. Select Issue Type** screen, select a new issue type (i.e. a standard issue type) and select **Next**.
4. If the sub-task's current status is not an allowed status for the new issue type, the **Step 2. Select New Status** screen is displayed. Select a new status and select **Next**.
5. In the **Step 3. Update Fields** screen, you will be prompted to enter any additional fields if they are required. Otherwise, you will see the message 'All fields will be updated automatically'. Select **Next**.
6. The **Step 4. Confirmation** screen is displayed. If you are satisfied with the new details for the issue, select **Finish**.
7. The issue will be displayed. You will see that it is no longer a sub-task, that is, there is no longer a parent issue number displayed at the top of the screen.

Converting an issue to a sub-task

1. Navigate to the issue you would like to convert.
2. Select **More > Convert to sub-task**.
3. In the **Step 1. Select Parent Issue and Sub-Task Type** screen, type or select the appropriate parent issue type and the new issue type (i.e. a sub-task issue type). Select **Next**.
4. If the issue's current status is not an allowed status for the new issue type, the **Step 2. Select New Status** screen is displayed. Select a new status and select **Next**.
5. In the **Step 3. Update Fields** screen, you will be prompted to enter any additional fields if they are required. Otherwise, you will see the message 'All fields will be updated automatically'. Select **Next**.
6. The **Step 4. Confirmation** screen is displayed. If you are satisfied with the new details for the issue, select **Finish**.
7. The issue will be displayed. You will see that it is now a sub-task, that is, its parent's issue number is now displayed at the top of the screen.

Note: You will not be able to convert an issue to a sub-task if the issue has sub-tasks of its own. You first need to convert the issue's sub-tasks to standalone issues; you can then convert them to sub-tasks of another issue if you wish. Sub-tasks cannot be moved directly from one issue to another you will need to convert them to standard issues, then to sub-tasks of their new parent issue.

Restricting access to an issue

When creating (or editing) an issue, you can restrict access to that issue to members of your team who are part of a chosen security level. To be able to set the security level for an issue, your administrator must add you to the appropriate issue security level, and also grant you the 'Set Issue Security' permission for the appropriate projects.

1. Create/edit the relevant issue.

2. In the **Security Level** drop-down field, select the desired security level for the issue. You will only see the security levels you belong to.
3. Save the issue. It is now only accessible to members of the specified security level. Users who are not members of this security level will not be able to access that issue, or see it in any filters, queries, or statistics.

Creating issues using the CSV importer

If you have the [Create Issue project permission](#) and the [Bulk Change global permission](#) for the relevant projects, you can create issues in bulk by using a comma-separated value (CSV) file. To find out what permissions you have, contact your Jira admin.

CSV files are text files that represent tabulated data and are supported by most applications that handle tabulated data, such as Microsoft Excel, Numbers, and databases.

Jira's CSV importer (also CSV file import wizard) allows you to import data from external systems that export their data in a tabulated format. Also, you can create your own CSV file to bulk create or update issues.

On this page:

- [Preparing your CSV file](#)
- [Running the CSV file import wizard](#)
- [Tips for importing CSV data into issue fields](#)

i Jira admins have access to more advanced features for importing issues. If you are planning to import a lot of issues from an external application, contact your Jira admin.

Read more about advanced import functionalities for Jira administrators in [Migrating from other issue trackers](#) and [Importing data from CSV](#)

There are two main steps to use the CSV importer:

1. [Preparing your CSV file](#)
2. [Running the CSV import wizard](#)

Preparing your CSV file

The CSV importer assumes that your CSV file is based off a default Microsoft Excel-styled CSV file:

- Fields are separated by commas
- Any content that must be treated literally, such as commas, new lines/ (e.g."carriage returns"), or angle brackets, are enclosed in quotes

i For Microsoft Excel and OpenOffice, you dont need to place cell values in quotation marks. These applications do this automatically.

CSV file requirements

In addition to being "well-formed", your CSV file should meet the following requirements:

- The file must have a header row. The CSV Importer uses the header row to determine how to map data from the CSV file's second row and beyond to Jira fields.
- The header row must contain the **Summary** column for the issues summary.
- The header row should not contain any punctuation, except for commas separating columns or fields. Otherwise, the import may work incorrectly.
- Commas as column or field separators can't be omitted throughout the file.

```
Summary, Assignee, Reporter, Issue Type, Description, Priority
"Test issue", admin, admin, 1, ,
```

```
Summary, Assignee, Reporter, Issue Type, Description, Priority
"Test issue", admin, admin, 1
```


Encapsulating Jira data structure in your CSV file

To import issues correctly, you should know how to encapsulate different data in your CSV file. In this way, Jira will be able to process them and create or update issues correctly.

Here, you'll find tips on and examples of building a CSV file with multiple-line fields, special characters, multiselect fields, and many more.

Learn more about the specifics of issue fields in [Tips for importing CSV data into issue fields](#).

Capturing data that spans multiple lines

Use double quotation marks (") in your CSV file to capture data that spans multiple lines. For example, during the import, Jira will treat the following as a valid CSV file with a single record:

```
Summary, Description, Status
"Login fails", "This is on
a new line", Open
```

Treating special characters literally

Put the text with special characters in double quotation marks (") to treat these characters literally. Once imported to Jira, these special characters will be stored as part of Jira's field value. Examples of special characters include carriage returns/enter characters, commas, etc.

If the text contains words or phrases in quotation marks and you want Jira to treat these quotation marks literally, put these words or phrases in another pair of quotation marks.

- Your CSV file may contain a value like: "Clicking the "Add" button results in a page not found error". Pay attention that the name of the button is put in two pairs of quotation marks.
- After the import, Jira will store this value as: Clicking the "Add" button results in a page not found error. The name of the button is stored in one pair of quotation marks, as it should be.

Aggregating multiple values into single issue fields

You can import multiple values into an issue field that accepts multiple values. For example, **Fix (for) Version, Affects Version, Component, Labels**. To do this, your CSV file must specify the same column name for each value you wish to aggregate into the mapped issue field. The number of column names must match the maximum number of values to be aggregated into the mapped field.

For example:

```
IssueType, Summary, FixVersion, FixVersion, FixVersion, Component, Component
bug, "First issue", v1, , , Component1,
bug, "Second issue", v2, , , Component1, Component2
bug, "Third issue", v1, v2, v3, Component1,
```

In this example, the **Component** field of the second issue and the **Fix Version** field of the third issue will generate multiple values in appropriate issue fields after the import.



Be aware that only a limited number of issue fields support multiple values. The CSV importer will not allow you to import aggregated data into issue fields that only support a single value.

Importing issues into multiple projects

You can import issues from your CSV file into different projects through a CSV file import. To do this:

- The CSV file must have two additional columns with the exact names of **Project Name** and **Project Key**. These are the mandatory values required for the correct import of issues from the CSV file to specific Jira projects.
- Ensure that every issue represented in your CSV file contains the appropriate name and key in these columns for the projects to which they will be imported.

i If you aren't a [Jira administrator](#), you won't be able to map the fields **Project Name** and **Project Key** to Jira's fields. You'll need to select different fields for mapping. Learn more about this in [Tips for importing CSV data into issue fields](#).

For example:

```
IssueType, Summary, Project Name, Project Key
bug, "First issue", Sample, SAMP
bug, "Second issue", Sample, SAMP
task, "Third issue", Example, EXAM
```

In this example, the first and second issues will be imported into the "Sample" project (with project key "SAMP") and the third issue will be imported into the "Example" project (with project key "EXAM"), assuming you match the "Project Name" and "Project Key" fields in your CSV file to the **Project name** and **Project key** issue fields respectively during the CSV file import wizard.

Importing work log entries

Your CSV file can contain work log entries. You need to use seconds to track the time spent. For example:

```
Summary,Worklog
Only time spent (one hour),3600
With a date and an author,2012-02-10 12:30:10;wseliga;120
With an additional comment,Testing took me 3 days;2012-02-10 12:30:10;wseliga;259200
```

i If you aren't a [Jira administrator](#), you won't be able to map the **Worklog** field to Jira's field. You'll need to select different fields for mapping. Learn more about this in [Tips for importing CSV data into issue fields](#).

Importing to multi select custom fields

Your CSV file can contain multiple entries for the one multi select custom field. Here's an example showing how to populate the multi select custom field with multiple values:

```
Summary,Multi Select,Multi Select,Multi Select
Sample issue,Value 1,Value 2,Value 3
```

Importing cascading choice custom fields

You can import values to a custom field with cascading choice by using the following syntax:

```
Summary, My Cascading Custom Field
Example Summary, Parent Value -> Child Value
```

The -> separator allows you to import the hierarchy.

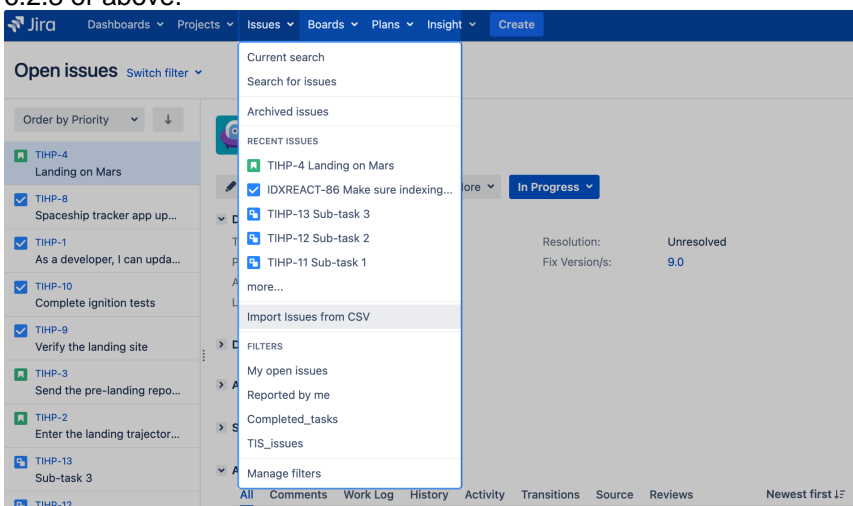
i Currently Jira does not support importing multi-level cascading select fields via CSV ([JRASERVER-34202](#) - Allow CSV import to support Multi-Level Cascading Select plugin fields **GATHERING INTEREST**).

Running the CSV file import wizard

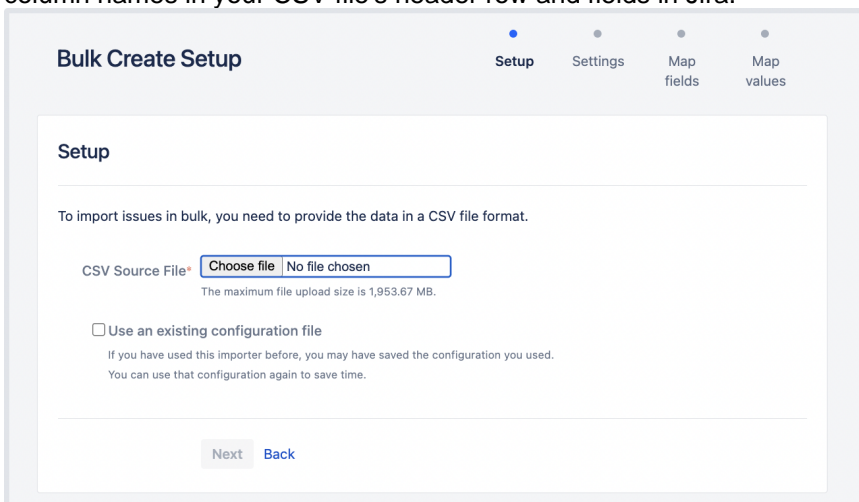
i If your Jira installation has existing data, you should [back it up](#) before proceeding with the following steps.

To import a CSV file with your Jira issues:

1. Select **Issues > Import Issues from CSV** to open the **Bulk Create Setup** page. If you do not have the option **Import issues from CSV**, your Jira admin must update the Jira Importers plugin to version 6.2.3 or above.



2. On the **Bulk Create Setup** page, select your **CSV Source File**. Leave the **Use an existing configuration file** checkbox cleared if you do not have a configuration file or if you want to create a new configuration file. Configuration files specify a mapping between column names in your CSV file's header row and fields in Jira.



i If you select **Use an existing configuration file** checkbox, you will be asked to upload an **Existing Configuration File**. If you don't select this option, Jira will automatically create a configuration file that you can use for further imports.

3. Select **Next** to proceed to the **Settings** step of the CSV file import wizard.
4. On the Settings page, populate the required fields and select **Next**.
 - **Import to Project:** select a project where you want to create or update issues.
 - **File encoding:** the type of character encoding in your CSV file.
 - **Delimiter:** a character that separates columns or fields in your CSV file. If your delimiter is a tab, specify it as `/t`.
 - **Date format:** the format of dates in your CSV file.

- On the **Map fields** page, map the fields from your CSV file to issue fields in the selected project. If you also want to map values of these fields, select the **Map field value** checkbox. After you finish, select **Next**.

CSV Field	Issue field	Map field value
Assignee	Issue Assignee	<input checked="" type="checkbox"/>
Created	Issue Created	<input checked="" type="checkbox"/>
Issue type	Issue type	<input checked="" type="checkbox"/>
Resolution	Issue Resolution	<input checked="" type="checkbox"/>
Summary	Issue Summary	<input checked="" type="checkbox"/>

i Make sure you map the **Summary** field. Its mandatory for creating or updating an issue.

The **Dont map this field** option means that you choose not to map a field from the CSV file to any issue field in the project. As a result, after the import, you wont see the unmapped fields in issues.

Use this option if:

- You cant find the right issue field for mapping. Learn more about this case in [Missing fields for mapping when importing issues through CSV in Jira](#).
- You dont have a Jira administrators permission to map a field. Learn more about such fields in [Tips for importing CSV data into issue fields](#).

- On the **Map values** page, you can check and reset the target values of the issue fields for which youve selected the **Map field value** checkbox on the previous page. If you havent selected the checkbox, proceed to the next step.


Example

Your issue types may have a CSV field value of **Feature Request**, which you may want to map to the issue type field value **Feature**.

- i**
- On this page, youll see all values of all fields for which youve selected the **Map field value** checkbox on the **Map fields** page.
 - If you want a field to be empty after the import, select **Import as blank** in **Target value in Jira**.

- If you havent chosen to map values for username-based fields like **Reporter** and **Assignee**, usernames from the file will be automatically mapped to existing usernames (lowercase) in Jira.
- If you import descriptions with line breaks, dont select the **Map field value** checkbox for the **Description** field. Otherwise, Jira may remove line breaks during the import.
- Regardless of whether or not you select the **Map field value** checkbox, Jira will automatically create usernames based on the data in your CSV file if they have not already been defined in Jira.

7. You may want to check your configuration for errors or warnings before running the import. To do it, select **Validate**. If your file contains warnings or errors, we recommend fixing them and uploading the file again. To learn more, download the detailed log of the validation.

 If you need to import another CSV file with the same (or similar) settings to what you used through this procedure, select the **save the configuration** link to download a CSV configuration file, which you can use at the first step of the CSV file import wizard. For example, you may want to use the same field or value mappings for the next imports.


8. When youre sure about the correctness of data in your CSV file, select **Begin Import**. The importer will display updates as the import progresses, then a success message when the import is complete. After the import is completed, you can check the created or updated issues in Jira. Also, you can download the detailed log of the import or save the configuration for future use.
9. Congratulations! You have successfully imported your CSV data into Jira! If you have any questions or encounter any problems, contact [Atlassian support](#).

Tips for importing CSV data into issue fields

Here are some helpful tips on importing data from your CSV file into specific issue fields:

Issue field	Import notes
Project	CSV data is imported on a per-project basis. You can either specify an existing projects as the target, or the importer will automatically create new projects during the import.
Summary	This is the only required field.
Component(s)	Import issues with multiple components by entering each component in a separate column.
Affects Version(s)	Import issues with multiple Affects Versions by entering each version in a separate column.
Fix Version(s)	Import issues with multiple Fix Versions by entering each version in a separate column.
Due Date	Make sure to use the date format specified on the second step of the CSV import wizard: dd /MMM/yy h:mm a.
Issue Type	If not specified in your CSV file, imported issues will be given the default (i.e. first) Issue Type as specified in your Jira system. For more information, see Defining issue type field values . You can also create new values on the fly during the import process.
Labels	Import issues with multiple labels by entering each label in a separate column or by putting all labels in one column, delimited by a space.
Priority	If not specified in your CSV file, imported issues will be given the default (i.e. first) Priority as specified in your Jira system. For more information, see Defining priority field values .

	You can also create new values on the fly during the import process.
Original Estimate	Set this value as a total number of seconds.
Remaining Estimate	Set this value as a total number of seconds.
Users	<p>Choose to automatically create Jira users for any values of the Assignee or Reporter field.</p> <ul style="list-style-type: none"> • Users will be created as active accounts in Jira. Users should receive emails with passwords for the first-time login to Jira. • To create full names for users with no real names, Jira will take the part of their email addresses before the @ character. • If you are using external user management system, Jira users won't be created during the import. Instead, the import wizard will give you a list of any new users that need to be created. You will need to create the users in your external user repository before running the import. • If you have a user-limited license (a personal license), and the number of users is larger than the limit, then the import will be stopped. You'll see the list of users that can't be created. • If Assignee and Reporter aren't mapped, no usernames will be created.
Other fields	<p>If you wish to import any other fields, you can choose to map them to one or multiple Jira custom fields. If your custom fields don't exist yet in Jira, the importer can automatically create them for you.</p> <p>If your custom field is a date field, please use the date format specified in the second step of the CSV import wizard: <code>dd/MMM/yy h:mm a</code>.</p>

 If you aren't a [Jira administrator](#), you won't be able to map the following fields in the CSV file to their equivalents in Jira:

- Attachments
- Comment Body
- Date Created
- Date Modified
- Date Resolved
- Project Key
- Project Name
- Project Type
- Resolution
- Status
- Worklog

To complete the field mapping, select either any other available field or the **Dont map this field** option.

Known issues

In case your issues get created but the multi-select fields are not populated, contact your admin. They can rerun the import by using Jira admin tools.

CSV import may run incorrectly for a project with only one issue type and a required custom field with context. For details and the workaround, see [JRASERVER-41584](#).

Editing and collaborating on issues

Resolve your customer requests more efficiently with these tips and tricks for editing and collaborating on Jira Service Management issues.

In addition to learning about the basics of editing and commenting on an issue, you can refer to this page for help with:

- Using the wiki toolbar to make your comments and descriptions pop
- Sharing issues with your team and adding request participants
- Keeping track of issues with labels and issue watchers


On this page:

- [Attaching files and screenshots](#)
- [Collaborating on issues](#)
- [Editing issue details](#)
- [Commenting on issues](#)
- [Canned responses for comments](#)
- [Formatting text with wiki markdown](#)
- [Tracking issues with labels](#)
- [Watching and voting for issues](#)
- [Reordering sub-tasks on an issue](#)

Attaching files and screenshots

If your administrator has enabled file attachments, you and your customers can attach files and screenshots to issues you're working on. See [Attaching files and screenshots to issues](#) for more information.

Collaborating on issues

You can easily keep your team informed by using the **Share** () button to share an issue with other Jira users. If your administrator has enabled anonymous access, you can also share issues by entering the email address of a non-Jira user.

If you want to invite members of your team to help you work on an issue, you can mention them by typing @ and their username in the issue description or comment. People already involved in the issue, like the reporter or a commenter, will be listed first in the user list so you can select them faster. Note that the users you mention will be notified once you save the issue description or comment. In Jira Service Management, your administrator can also enable Request participants, which will appear as another issue field. You can add other agents and customers from your service desk project to help you resolve the original customer's request.

Editing issue details

What permissions do you need?

To edit an issue, you need the **Edit Issue** project permission for the issue's relevant project. If you do not have this permission, please contact your administrator.

To edit an existing issue, select **Edit** to open the **Edit Issue** dialog box and modify the issue details. If you want to change the fields you need to edit, select **Configure Fields > Custom** and choose the fields you want to show or hide. Select **Update** to save your changes.

Commenting on issues

What permissions do you need?

To add comments to an issue, i.e. to see the **Comment** button, you must have both of the following project permissions for the issue's relevant project:

- **Browse Project** permission to view the issue to be commented on
- **Add Comments** permission to add a comment to the issue.

Note that you automatically become a watcher of the issues that you comment on. You can disable this via the **Preferences > Auto watch** option in your profile.

What	How
Add a comment	Choose Click to add comment and type your comment for the issue.
Delete a comment	Select Delete at the bottom of the comment. Confirm that you want to remove this comment from the issue by selecting Delete when prompted.
Edit a comment	Select Edit at the bottom of the comment, and edit the text or restrictions (Viewable by...) as needed. When you save your revised comment, you'll see 'edited' displayed to indicate that the comment has been edited. You can hover over 'edited' to see who edited the comment and when.
Link to a comment	Right-click on the comment timestamp, and then copy the link address. Paste the copied permanent link into your email or chat message. Clicking the permanent link takes you to that particular comment in the Jira issue. If your Jira issue contains an extensive list of comments, the issue page will automatically be scrolled down so that the linked comment is visible.
Restrict a comment	Select Comment internally (for other internal agents and collaborators) or Share with customer (for customers) tab.

Canned responses for comments

Canned responses allow you to create, edit and manage responses that you can then use at any time, directly from the view issue screen. All agents in your project have access to the canned responses saved, so your colleagues can take advantage of your saved responses, and you can take advantage of theirs. Some typical examples of canned responses could be:


- a request for more information from the customer,
- an short message confirming work in ongoing and the next update will be in 24 hours, or
- a lengthy response asking a customer to accept terms and agreements before you continue with the work.

You can access your canned responses when adding a comment by selecting the canned responses icon



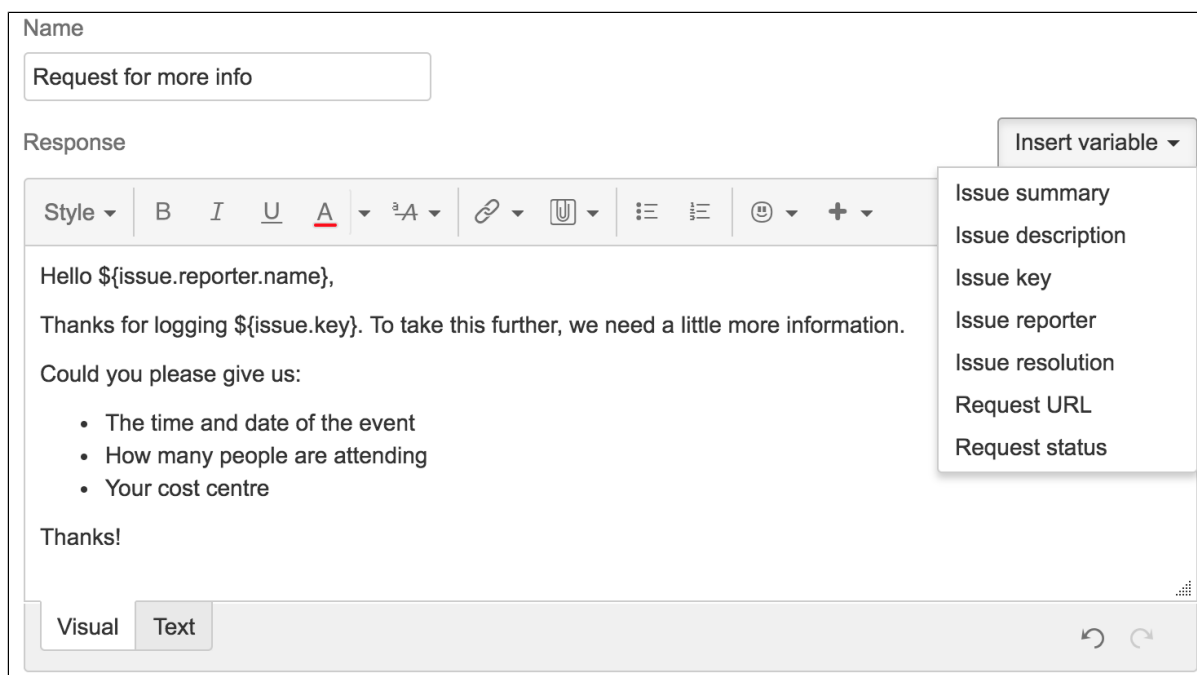
. You can add a response directly from the **Popular** or **Frequently** used menu, or you can search for your response by typing the name in the search field, and selecting it. The canned response will be added to the comment field. Feel free to further edit your comment, it won't be added to the issue until you add the comment.

To edit a canned response, select  > **Edit responses**, select the response you'd like to change and click **Edit**.

To add a canned response, you can either start with the text you've already added to the comment field, or with a blank comment field. Select  > **Save as a response**, and complete the form with a name for your response, the response, and then select **Save**.

Using variables in canned responses

A variable is a powerful way to customize your response. Using a variable in your response allows the response to automatically contain information related to the issue. For example, you may want to personalize your response by adding the reporter's name, or by adding the issue key. The current list of variables are shown on the image below:



The screenshot shows the Jira canned response editor interface. At the top, there is a 'Name' field containing 'Request for more info'. Below it is the 'Response' field, which is a rich-text editor. The editor has a toolbar with options for style (Style), bold (B), italic (I), underline (U), text color (A), background color (A), link, unlink, bulleted list, numbered list, emoji, and a plus sign for more options. The text in the editor reads: 'Hello \${issue.reporter.name}, Thanks for logging \${issue.key}. To take this further, we need a little more information. Could you please give us:

- The time and date of the event
- How many people are attending
- Your cost centre

 Thanks!'. A dropdown menu titled 'Insert variable' is open, showing a list of variables: Issue summary, Issue description, Issue key, Issue reporter, Issue resolution, Request URL, and Request status. At the bottom of the editor, there are tabs for 'Visual' and 'Text', and a refresh button.

Formatting text with wiki markdown

Jira application [Text Formatting Notation](#) allows you to use rich-text features, such as:

- Italic, bold, underlined text
- Multiple levels of headings
- Bullets, numbered lists, tables, and quotations
- Images
- Macros

When you edit an issue description, comment, or any rich-text field, you can expand the simple wiki editor toolbar to format your text and select **preview** to see how your formatted text will appear. Note that your Jira administrator can enable, disable and configure the renderer which allows you to use wiki markdown, so your options may vary slightly.

HTML macro

When using the HTML macro, which allow you to add HTML code to an issue, you should only use formatting as if you are including something inside the `{{<body>}}` directly. This prevents you from accidentally breaking the page formatting, or overriding Jira's CSS.

Note that if you're administrator has enabled the rich text editor, you'll still be able to format your content using wiki markdown, but if you select the [visual editor](#), you'll see the markdown applied directly.

Tracking issues with labels

Labeling helps you categorize and search for an issue. When viewing an issue, select **More > Labels** to add or remove labels, which will appear in the Details section:

▸ Details

Type: <input checked="" type="checkbox"/> Task	Status: IN PROGRESS (View Workflow)
Priority: = Medium	Resolution: Unresolved
Affects Version/s: None	Fix Version/s: 7.5
Component/s: Engine	
Labels: DOC	

You can click a label (e.g. **doc** in the above screenshot) to jump to the Issue Navigator and see a list of all issues that have this label. You can also add the [Labels Gadget](#) to your dashboard to quickly find issues with labels relevant to you and your team.

Watching and voting for issues

i What permissions do you need?

To view other users watching or voting for an issue, you need the **View Voters and Watchers** and **Manage Watcher List** project permissions.

If your administrator has set up the needed notification scheme, you can select **Start watching this issue** to be automatically notified of issue updates. You can also click the number of watchers on the issue to add other Jira users as watchers.

If your administrator has enabled the voting on issues, you can select **Vote for this issue** to encourage the responsible team to resolve or complete the issue.

Reordering sub-tasks on an issue

If you've added sub-tasks to an issue, and need to reorder them, you can drag and drop them on the issue navigator view of the parent issue. If you're using a board in Jira Software, you can also reorder the sub-tasks on the board view. However, these two methods are **independent** of each other. Reordering sub-tasks on the parent issue **will not** reorder tasks on an existing board, and vice-versa.

Linking issues

Issue linking allows you to create an association between two existing issues on either the same or different Jira servers. For example:

- An issue may *relate to* another.
- An issue may *duplicate* another.
- An issue may *block* another.

Issue linking also allows you to:

- Create a new linked issue from an existing issue in a service desk or business project.
- Create an association between an issue and a Confluence page.
- Link an issue to any other web page.

Your Jira administrator can customize the types of links that you can create, see [configuring issue linking](#).

On this page:

- [Creating a link to another issue on the same Jira site](#)
- [Creating a link to an issue on another Jira site](#)
- [Create a new linked issue from an existing issue in a service desk or business project](#)
- [Creating a link to a Confluence page](#)
- [Creating a link to any web page URL](#)
- [Deleting a link](#)
- [Searching for linked issues](#)

Issue links within an issue look like this:



Note: Resolved issues (i.e. issues with a Resolution set) are displayed in strike-through font, e.g. ~~DEMO-1~~.

i To create links on issues, you need to have the Link Issues permission in the project(s) to which the issues belong.

Creating a link to another issue on the same Jira site

1. Open the issue you wish to link to another issue in the same Jira site.

2. Select **More > Link** to display the **Link** dialog box.

3. Ensure that the **Jira Issue** item is selected at the left of the dialog box and then choose the type of link to be created from the **This issue** drop-down list.
 - ⓘ If your Jira system administrator has configured *fully reciprocal application links* between your Jira site and another one, a **Server** drop-down list may appear above the **This issue** list. If this is the case, ensure your Jira site appears or has been selected from the **Server** list.
4. In the **Issues** field, specify the issue(s) to be linked to your currently viewed/selected issue. There are two ways to do this:
 - Type the full issue key (e.g. **ABC-123**) or to link to multiple issues, press the 'Enter' key between each typed issue key.
 - ⓘ If you have previously browsed an issue, you can quickly find the issue by typing the first few letters of the issue key (or part of the Summary), which will appear in an 'autocomplete' drop-down list for selection:
 - OR:**
 - Click the **search for an issue** link to use the **Find Jira issues** popup, which allows you to perform either a simple **text search** or an **advanced search** for issues.
5. Optional: Add a **Comment** to describe why you are linking these issues.
6. Click the **Link** button at the bottom of the dialog.

Creating a link to an issue on another Jira site

⚠ To create this type of link, your Jira system administrator should have configured *fully reciprocal application links* between your Jira site and the other Jira site containing the issue(s) you want to link to.

1. Open the issue you wish to link to another issue.
2. Select **More > Link** to display the **Link** dialog box.
3. Ensure that the **Jira Issue** item is selected at the left of the dialog box.
 - ⓘ **Note:**
 - This option will not be available if your Jira system administrator has not configured an application link between your Jira site and the remote Jira site.
 - If, after selecting this option, you are prompted for authorization, you may be required to log in to the remote Jira site, which will allow your Jira site to access the remote Jira site *on behalf of your account on the remote Jira site*.
 - ⓘ This behavior means the application links configured between your Jira site and the remote Jira site use OAuth authentication.
4. If your Jira site is connected to multiple remote Jira sites, choose the relevant Jira site from the **Server** drop-down list.

5. Choose the type of link to be created from the **This issue** drop-down list.
6. Type the **Issue** key of the issue on the remote Jira site that you want to link to. Alternatively, you can search for issues on the remote Jira site by clicking the **search for an issue** link, which opens the **Find Jira issues** popup.
 - You can link to any issue on the remote Jira site to which you have access on that site.
7. Select the **Create reciprocal link** checkbox to create the complementary link on the remote issue you are linking to, back to your issue. For example, if you create a **blocks** link type to a remote issue, the reciprocal link generated on the remote issue will be **is blocked by** link type back to your local issue.
8. Optional: Add a **Comment** to describe why you are linking these issues.
9. Click the **Link** button at the bottom of the dialog.

Troubleshooting

Problem: If you selected the **Create reciprocal link** checkbox, but after clicking the **Link** button, you discover that a reciprocal link from the remote issue back to your issue has not been created, then your Jira system administrator has most likely created only a one-way link from your Jira site to the remote Jira site.

Solution: Ask your Jira system administrator to configure *fully reciprocal application links* between your Jira site and the remote Jira site.

Problem: If you attempted to create a reciprocal link but received the following message:

'A reciprocal link from issue 'XYZ-123' back to this issue was not created as the remote Jira server returned the following error: No Link Issue Permission for issue 'XYZ-123'.' (where 'XYZ-123' is the issue key on the remote Jira site),

then a reciprocal link on the remote Jira site will not have been created, because the user account through which you authenticated on the remote Jira site (at step 3 above) does not have the **Link Issues** project permission.

Solution:

- Ask the Jira project administrator(s) on the remote Jira site to grant your user account the **Link Issues** project permission for the relevant project(s) to which you need to create issue links.
- Alternatively, if the application link between your Jira site and the remote Jira site use OAuth authentication and you suspect you may have authenticated on the remote site with another user account that does not have the **Link Issues** project permission, repeat the procedure above but during the authorization step (at step 3), authenticate on the remote site with a user account which has this permission.
 - If you are not prompted for authentication during authorization, try clearing your browser's cookies first and repeat the procedure again.

Create a new linked issue from an existing issue in a service desk or business project

i To create a linked issue, you need to have **Create issue** and **Linked Issues** permissions in the destination project(s).

To create a linked issue:

1. Open the issue from which you wish to create the linked Jira issue.

- In the Issue screen, select **More > Create linked issue** to display the **Create Linked Issue** dialog box. The newly created linked issue contains the same Project, Issue Type, and Summary information stored in the original issue. It is also linked to the service desk issue, in this case CTF-2.

The screenshot shows the 'Create linked issue' dialog box. It has the following fields and options:

- Project:** Charlie Travel Sydney (CTS)
- Issue Type:** Problem
- Created issue:** causes
- Linked issues:** CTF-2
- Summary:** Fix software bug in app XYZ
- Description:** The print dialog is missing the orientation features. Unable to print to landscape.

At the bottom right, there are 'Create' and 'Cancel' buttons.

- Select the destination **Project** in which the new linked issue is to be created.
- Select the correct Issue Type for the new linked issue.
- In the **Linked issues** field, specify issue(s) to be linked to your new linked issue.
- Edit the linked issue **Summary**.
- Edit the **Description** and describe why you are linking these issues.
- Select the **Copy attachments** checkbox to include any attachments from the original issue.
- Select the **Copy links** checkbox to include any URLs from the original issue.
- Click the **Create** button at the bottom of the dialog.

Your linked issue has now been created.

Creating a link to a Confluence page

! This feature is only supported in Confluence versions 4.0 or later.

! To create this type of link, your Jira system administrator needs to have configured an *application link* between your Jira site and the Confluence site containing the pages you want to link to.

- Open the issue you wish to link to another issue.
- Select **More > Link** to display the **Link** dialog box.
- Click the **Confluence Page** option at the left of the dialog box.
 - This option is not available if your Jira system administrator has not configured an application link between your Jira site and Confluence site.
- If more than one application link has been configured between your Jira site and other Confluence sites, then choose the appropriate Confluence site from the **Server** drop-down list.
- Specify the Confluence page to be linked to your currently viewed issue. There are two ways to do this:
 - In the **Page URL** field, enter the URL of a page on the Confluence site you want to link to. For example:

```
http://<confluence-server>/display/ds/Welcome+to+the+Confluence+Demonstration+Space
```

- Click the **search for a page** link. The **Link** dialog box is replaced by the **Find a Confluence page** dialog box.
 - If you are prompted for authorization, you may be required to log in to the Confluence site, which will allow your Jira site to access the Confluence site *on behalf of your account on the Confluence site*. This behavior means the application links configured between your Jira site and the remote Confluence site use OAuth authentication.
 - a. In the first **Search** field, specify one or more search terms that appear in the page you want to link to. This field is mandatory.
 - b. Optional: In the second **Search** field, select the Confluence space to further narrow down the search.
 - c. Click the **Search** button and then the title of the page you want to link to.
- 6. Optional: Add a **Comment** to describe why you are linking these issues.
- 7. Click the **Link** button at the bottom of the dialog.

Troubleshooting

✘ **Problem:** If Confluence page links you create show **Failed to load** on the issue or if you attempted to search for a Confluence page but received the following message:

'Content on the Confluence site could not be accessed because the Confluence server's 'Remote API' feature is disabled. The Confluence system administrator must enable this 'Remote API' feature for Jira to successfully access this content.'

then Jira was unable to communicate with the Confluence server to either:

- retrieve information about the link or
- conduct a Confluence page search in the **Find a Confluence page** dialog box.

✔ **Solution:**

Ask the Confluence system administrator to enable the **Remote API (XML-RPC & SOAP)** feature, since this Confluence feature is disabled by default. See [Enabling the Remote API](#) in the Confluence documentation for details.

Creating a link to any web page URL

1. Open the issue you wish to link to another issue.
2. Select **More > Link** to display the **Link** dialog box.
3. Click the **Web Link** option at the left of the dialog box.
4. Specify the **URL** of the web page you want to link to.
5. Specify the **Link Text** that will appear in the **Issue Links** section of the 'view issue' page and will be hyperlinked to your URL.
6. Optional: Add a **Comment** to describe why you are linking these issues.
7. Click the **Link** button at the bottom of the dialog.

Deleting a link

1. Go to an issue that contains links, and locate the **Issue Links** section.
2. Hover your mouse over the link you wish to delete, and click the **Delete** (trashcan) icon that appears.

Searching for linked issues

You can search for issues that are linked to a particular issue. See [Advanced searching](#) for more information.

• Be aware that this functionality does not extend to issues on a remote Jira server.

Editing multiple issues at the same time

At some point, you may need to change multiple issues at the same time. You can do this by performing a Bulk Change operation.

There are restrictions placed on some of the bulk operations. For example, if you select multiple issues with different workflows, you can only transition them in groups with the same workflow, and one group at a time. You can only bulk change 1000 issues as loading more than that might result in the Out Of Memory error. The restrictions are explained further in the relevant sections.

On this page:

- [Before you begin](#)
- [Delete multiple issues](#)
- [Move multiple issues](#)
- [Edit multiple issues](#)
- [Watch / stop watching multiple issues](#)
- [Archive multiple issues](#)

Before you begin

Required permissions

To perform a bulk change operation, you need the appropriate project-specific permission and the global Bulk Change permission. For example, you would need to have both the **Move Issue** and **Bulk Change** permissions to perform the **Bulk Move** operation. If you cannot access the Bulk Change functionality, ask your project admin to grant you the required permissions.

Using the bulk change wizard

The bulk change wizard will progress you through your bulk change. To open the wizard:

- Perform a search with the required filters to produce a list of issues and select **Tools > Bulk Change**.
- Bulk edit issues in the Backlog. Use Shift+select to choose multiple issues and then right-click the selected items to start the bulk change operation.

Bulk Operation

- Choose Issues
Selected 1 issues from 1 project(s)
- **Choose Operation**
- Operation Details
- Confirmation

Step 2 of 4: Choose Operation


Choose the operation you wish to perform on the selected 1 issue(s).

<input type="radio"/>	Edit Issues	Edit field values of issues
<input type="radio"/>	Move Issues	Move issues to new projects and issue types
<input type="radio"/>	Transition Issues	Transition issues through workflow
<input type="radio"/>	Delete Issues	Permanently delete issues from Jira
N/A	Archive Issues	NOTE: You might not have the permission to archive one or more issues, or you're trying to archive subtasks instead of issues.
<input type="radio"/>	Watch Issues	Watch all the selected issues. You will receive notifications when any of these issues are updated.
<input type="radio"/>	Stop Watching Issues	Stop watching all the selected issues. You will no longer receive notifications when any of these issues are updated.

To go back to any step of the operation, select the relevant step in the menu on the left-hand side of the wizard. Selecting **Cancel** will cancel the entire process.

Disable notifications for bulk operations

You can disable mail notifications for a particular bulk operation by deselecting the **Send Notification** checkbox in the bulk operation wizard. For this option to be available, you must be a [Jira administrator](#) or [project administrator](#) of all the projects associated with your selected issues. Deselecting **Send Notification** only disables Jira notifications. It doesn't affect notifications that are sent to your service desk customers.

i **Disable customer notifications.** Some bulk operations, such as **Change comment**, might trigger email notifications to your customers. To prevent a flurry of emails, a Jira admin can temporarily disable outgoing mail in **Administration**  **>System>Mail > Outgoing mail**. This setting controls both Jira and customer notifications, so remember to turn it back on when you're done with your bulk edit.

Transition multiple issues

This bulk operation allows you to transition multiple issues through a workflow at the same time. You can only perform one transition bulk operation at a time. You will also need to provide any values required to complete the transition. For example, to close multiple issues, you will need to provide a value for the Resolution field, such as Done, Fixed, or Won't Fix.

To transition multiple issues:

1. Perform a search with the required filters to produce a list of issues.
2. Select **Tools > Bulk Change**.
3. Select the issues you'd like to perform the bulk operation on and then select **Next**.
4. Select **Transition Issues** and then select **Next**.
5. Select the available workflow action. The actions available are dependent on the issues (and their associated workflows) that you have selected. Select **Next**.
6. Select a value for any required fields for this transition, and if available, decide whether you'd like to send email notifications. Select **Next**.
7. Review your bulk operation, and select **Confirm** when you are happy with the operation.

Delete multiple issues

This bulk operation allows you to delete multiple issues at the same time. To delete multiple issues:

1. Perform a search with the required filters to produce a list of issues.
2. Select **Tools > Bulk Change**.
3. Select the issues you'd like to perform the bulk operation on, and select **Next**.
4. Select **Delete Issues** and then select **Next**.
5. If available, decide whether you'd like to send email notifications. Select **Next**.
6. Review your bulk operation, and select **Confirm** when you are happy with the operation.

Move multiple issues

This bulk operation allows you to move multiple issues at the same time. The issues you're moving need to be mapped to both a project and an issue type, and in doing this, you may need to also map the status and fields of the issues. Subtasks need to be mapped, too.

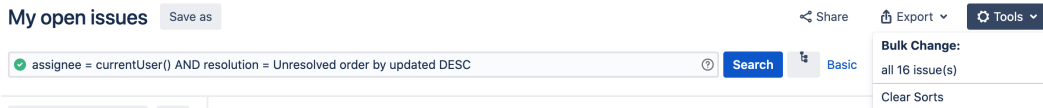
You can bulk move both standard issues and sub-tasks to another project and issue type, as well as convert a sub-task to an issue and vice versa. To learn more about issue types, see [Issue fields and statuses](#).

You can also select both a sub-task and its parent to bulk move. However, you'll only be asked to move sub-tasks if you're moving a parent issue to another project. This is to maintain the parent/sub-task relationship (i.e. the sub-task is always located in the same project as the parent issue). For example, you have issue B being a sub-task of issue A and you try to bulk move both A and B simultaneously. You will be prompted to select a target project and issue type for issue A. If you select a new project for A, you will be prompted to move the sub-task to a new issue type based on issue A's new project. If you don't change the project for issue A, the sub-task will not be required to be moved.

To move multiple issues:

1. Perform a search with the required filters to produce a list of issues.

2. Select **Tools** > **Bulk Change**.



3. Select the issues you'd like to perform the bulk operation on, and select **Next**. The bulk move operation may require additional information depending on which issues you have selected to move (expand the following dropdown for details).

<p>Select Projects and Issue Types</p>	<p>The first step of the Bulk Move wizard is to choose which projects and issue types you will move your issues to. The target project and issue type will determine whether extra steps will be required to migrate statuses and fields.</p> <p>Selected issues are grouped by their current project and issue type. You can either select a new project and issue type for each one or choose to move all standard issues to a single project and issue type.</p>
<p>Select Projects and Issue Types for Sub-Tasks</p>	<p>If you are moving issues with sub-tasks to another project, you will also need to move the sub-tasks to the new project. You can also elect to change the issue types of the sub-tasks being moved if you need to.</p>
<p>Select status migration mappings for invalid statuses</p>	<p>As multiple workflows can be active simultaneously, some statuses associated with the collection of selected issues may not be valid in the target workflow. In this case, you should map invalid statuses to valid statuses in your new workflow.</p>
<p>Select values for required fields and fields with invalid values</p>	<p>To adhere to the field configuration scheme associated with the target project and issue type, it may be necessary to update/populate required fields (e.g. fields that are required in the target project, but may not have been in the original project). For each field that needs to be populated, you will be prompted to supply a value. This value will be applied to all issues that are being bulk moved together.</p> <p>For the following fields, you can select from a list of possible values provided for you:</p> <ul style="list-style-type: none"> • Component • Affects Version • Fix Version • Custom fields of type "Version-Picker" <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>i Versions which have been archived in the target project cannot be selected as the target when performing a bulk move. If you need to move issues into an archived version, you will need to first unarchive the version in the target project.</p> </div> <p>It is possible to retain original field values that are valid in the target destination by checking the Retain checkbox associated with the field. For example, some issues may already include a valid custom field value these values can be retained, while issues that require an update will adopt the value specified on the Field Update screen.</p> <ul style="list-style-type: none"> • Checked: the original value is retained where possible. The field will not be updated with the specified new value. • Unchecked: all fields will be updated with the specified new value. Note that the "Retain" checkbox is not available for the following fields, since an explicit mapping is required:

	<ul style="list-style-type: none"> • Component • Affects Version • Fix Version • Custom fields of type "Version-Picker"
--	---

4. Select **Move Issues** and then select **Next**.
5. Confirm your changes and complete the operation. Note that you'll need to specify "Status migration mappings for invalid statuses" and "Values for required fields and fields with invalid values" once for each different target project and issue type combination. When all move parameters e.g. target project, status mappings and field updates have been specified for all issues, you will be presented with a confirmation screen displaying all changes that will be made to the issues being moved. The following details are displayed as applicable:

- **Issue Targets:** the target project and issue type
- **Workflow:** the target workflow and invalid status mappings
- **Updated Fields:** new values for fields that require updating
- **Removed Fields:** values to be removed in fields that are not valid in the target

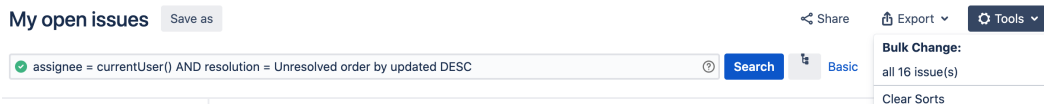
The issues will only be moved once the **Confirm** button is clicked from the confirmation page. If the operation is exited anytime before this step, no changes will be made to the issues

Edit multiple issues

This bulk operation allows you to edit multiple issues at the same time. The bulk edit operations available depend on the issues selected and the nature of the field/s you want to change.

To edit multiple issues:

1. Perform a search with the required filters to produce a list of issues.
2. Select **Tools > Bulk Change**.



3. Select the issues you'd like to perform the bulk operation on and then select **Next**.
4. Select **Edit Issues** and then select **Next**.
5. Select the bulk edit operation from the list of available operations (expand more information for a full list of available and unavailable operations, and their conditions).

Available Operations	Conditions
Change Affects Version/s	<ul style="list-style-type: none"> • Selected issues belong to one project, and that project has version/s • This field is not hidden in any field configurations the selected issues belong to • Current user has "edit issue" permission for all the selected issues
Change Assign To	<ul style="list-style-type: none"> • This field is not hidden in any field configurations the selected issues belong to • Current user has "assign issue" permission for all the selected issues
Change Comment	<ul style="list-style-type: none"> • This field is not hidden in any field configurations the selected issues belong to • Current user has "comment issue" permission for all the selected issues
Change Component/s	<ul style="list-style-type: none"> • Selected issues belong to one project, and that project has component/s • This field is not hidden in any field configurations the selected issues belong to • Current user has "edit issue" permission for all the selected issues
Change Due Date	<ul style="list-style-type: none"> • This field is not hidden in any field configurations the selected issues belong to

	<ul style="list-style-type: none"> • Current user has "edit issue" permission for all the selected issues • Current user has "schedule issue" permission for all the selected issues
Change Fix For Version/s	<ul style="list-style-type: none"> • Selected issues belong to one project, and that project has version/s • This field is not hidden in any field configurations the selected issues belong to • Current user has "edit issue" permission for all the selected issues
Change Issue Type	<ul style="list-style-type: none"> • Current user has "edit issue" permission for all the selected issues
Change Priority	<ul style="list-style-type: none"> • This field is not hidden in any field configurations the selected issues belong to • Current user has "edit issue" permission for all the selected issues
Change Reporter	<ul style="list-style-type: none"> • This field is not hidden in any field configurations the selected issues belong to • Current user has "edit issue" permission for all the selected issues • Current user has "modify reporter" permission for all the selected issues
Change Security Level	<ul style="list-style-type: none"> • This field is not hidden in any field configurations the selected issues belong to • All the selected projects are assigned the same issue level security scheme • Current user has "edit issue" permission for all the selected issues • Current user has "set issue security" permission for all the selected issues
Change Custom Fields	<p>The "Change Custom Fields" operation is available only if:</p> <ul style="list-style-type: none"> • a global custom field exists OR • an issue type custom field exists and the issues are all of this specific issue type OR • a project custom field exists and the issues are all of the same project
Edit a Closed Issue	<ul style="list-style-type: none"> • Your workflow must allow editing of closed issues
Change Sprint	<p>You need to specify the sprint ID.</p> <ul style="list-style-type: none"> • This operation only affects active and future sprints, i.e. closed/completed sprints are not included when bulk editing the Sprint field.

The fields listed in this section have no operations for bulk editing. This is because there is an alternative method or it is not logical to perform bulk edit on them.

The following system fields are unavailable for bulk editing:

- Attachments
- Summary
- Description
- Environment
- Project. Use "Bulk Move" to move issues between projects
- Resolution. Use "Bulk Workflow Transitions" to modify the resolution of issues
- Time Tracking fields: Original Estimate, Remaining Estimate, Time Spent

The following custom field types are unavailable for bulk editing:

- Import Id
 - Read Only Text
- Select a value for any required fields for this operation, and if available, decide whether you'd like to send email notifications. Select **Next**.
 - Review your bulk operation, and select **Confirm** when you are happy with the operation.

Watch / stop watching multiple issues

These bulk operations allows you to start watching or stop watching multiple issues at the same time.

To watch multiple issues:

1. Perform a search with the required filters to produce a list of issues.
2. Select **Tools > Bulk Change**.
3. Select the issues you'd like to perform the bulk operation on and then select **Next**.
4. Select **Watch Issues** and then select **Next**.
5. Review your bulk operation, and select **Confirm** when you are happy with the operation.

To stop watching multiple issues:

1. Perform a search with the required filters to produce a list of issues.
2. Select **Tools > Bulk Change**.
3. Select the issues you'd like to perform the bulk operation on, and select **Next**.
4. Select **Stop Watching Issues** and then select **Next**.
5. Review your bulk operation, and select **Confirm** when you are happy with the operation.

Archive multiple issues

 This functionality is available with the **Jira Data Center** license.

If you want to archive thousands of issues at once, instead of selecting the issues manually, you can make a bulk change. By default, this option allows you to archive all the issues on the current page or a maximum of 1000 issues. However, a system admin can raise this limit if needed.

The issues will be archived with all their subtasks. Note that issues might have multiple subtasks so archiving might take a while because of a great number of subtasks.

To archive multiple issues:

1. Perform a search with the required filters to produce a list of issues. You can archive all the issues on the current page or a maximum of 1000 issues.
2. Select **Tools > Bulk Change**.
3. Select the issues you'd like to perform the bulk operation on, and select **Next**.
4. Select **Archive issues** and then select **Next**.
5. Review your bulk operation, and select **Confirm** when you are happy with the operation.


Moving an issue

Sometimes, an issue may belong to a different project, and you may want to move this issue to another project. You can easily do this by using the **Move issue** wizard.

Before you begin:

- You must have the **Move Issues** permission for the project that has the issue that you want to move.
- You must have the **Create Issues** permission for the project that you wish to move your issue to.

If you do not have either of these permissions, please contact your Jira administrator to have these added to your user profile.

 If you wish to move multiple issues between projects at the same time, please refer to the documentation on [bulk moving issues](#).

Moving an issue

The **Move issue** wizard allows you to specify another project in your Jira instance to move your selected issue to. As there may be significant differences in the configuration of your original project and target project, the **Move issue** wizard allows you to change certain attributes of the issue. These include:

- **Issue type** If your issue is a custom issue type that does not exist in your target project, you must select a new issue type. You can also choose to arbitrarily change the issue type.
- **Issue status** You may have set up custom issue statuses as part of a workflow. If you have assigned a custom status to your issue, and it does not exist in your target project, you must select a new issue status for your issue. You cannot arbitrarily change the issue status, i.e. the option to change the issue status will only appear if you are required to change it.
- **Custom fields** If you have defined **required** custom fields for your issue that do not exist in your target project, you must set values for them. You will only be prompted to enter the values for **required custom fields** in the target project that are missing values. If the custom fields of your original project also exist in your target project, and these custom fields are not required in the target project, you may need to set values for them, to move the issue successfully. If you wish to change the existing values for other fields on your issue, you can do this after the move is complete.

To move an issue:

1. View the issue that you wish to move.
2. Select **More > Move**.
3. The first page of the **Move issue** wizard is displayed. Complete the steps required.
4. The confirmation page will display with all of your changes. If you wish to revise any of your changes, you can select the appropriate step in the left-hand menu to return to that page of the wizard. Once you are happy with your changes, select **Move** to move the issue to the target project.
5. Your issue will be moved to the target project and displayed on screen. You can now edit the issue to make further changes, if you wish.

Moving related issues

- If your issue has sub-tasks, the **Move issue** wizard will also move the sub-tasks to the target project.
- If you're moving an epic, the **Move issue** wizard will not move the issues in the epic. The epic and the issues in the epic will still be linked to each other, but the issues in the epic will remain in the original project. You will need to move them separately.
- If your issue has values in the **Components** field, since the **components** field is project-tied, the target project must have the same component value already in place with the exact same name. Otherwise, the value will be lost during the move operation.

Troubleshooting

- Restricted comments appear to be removed after moving the issue. See this article: [Restricted comments disappear after moving an issue to a new project](#).

Visual editing

Formatting content in Visual mode gives you a What You See Is What You Get (WYSIWYG) experience. Formatting appears as you apply it, and you no longer have to flip to a Preview to see what your content will look like when saved. You still have the option to view the wiki markup by selecting the Text tab. You'll know you have access to the visual editor because you'll see the Visual and Text tabs.

The image shows two panels illustrating the visual editor. The top panel, labeled 'Visual', displays a rich-text toolbar with options for style, bold, italic, underline, text color, background color, link, unlink, list, and emoji. The content below the toolbar is rendered in a visual format: 'The new visual editor is **AWESOME!!**', 'I can apply formatting using the toolbar or typing wiki markdown, and it appears as I apply it! Things like', a bulleted list with items 'color', 'underline', and 'emoticons 😊', and 'and I even get'. Below this is a table with the header 'Tables' and the text 'that look like tables!'. At the bottom, there are tabs for 'Visual' (selected) and 'Text', and a lock icon with the text 'Viewable by All Users'.

The bottom panel, labeled 'Text', shows the same toolbar but with the 'Text' tab selected. The content is displayed as raw wiki markup: 'The new visual editor is `_*AWESOME*_!!`', 'I can apply formatting using the toolbar or typing wiki ma', '* `{color:#d04437}color{color}`', '* `+underline+`', '* `emoticons :)`', 'and I even get', '||Tables||', and '|that look like tables!|'. At the bottom, the 'Text' tab is selected, and the 'Visual' tab is visible but not active.

In Visual mode, you can still enter wiki markup syntax as you add your content, and it'll be rendered exactly as it'll display when you save. You can even flip between modes to view the formatted content, and the wiki markup syntax. You can also use the toolbar to format and style your content.

As Visual editing is really a preview of what we're working on, there's a few things that may not work quite as you'd expect them:


- Formatting content in a complex way can affect its ability to be rendered, things like tables in the cells of other tables, and adding images to table cells won't work.
- Pasting content may not work as expected, as the source content may really be formatted using a method we don't support. So pasting tables may work, and it may not, depending on the source. Pasting plain text is absolutely fine.
- If you have macros provided by 3rd party apps, and they're incompatible with Visual mode, you won't be able to edit the macro header, and its content will be rendered as text (wiki markup).

Scheduling an issue

You can schedule issue due dates in Jira Service Management to help your agents prioritize incoming customer requests and find overdue issues that need urgent attention. The powerful scheduling feature allows you to perform fixed and relative date searches based on specific due dates as well as arbitrary search periods. You can also perform advanced searches using Jira Query Language.

Scheduling an issue

To schedule an issue, populate its **Due** date field. This can be done either when creating an issue, or at a later stage by editing the issue.

 To enable Issue Scheduling, at least one group or project role must be given the Schedule Issues permission by your Jira administrator. Only users with the Schedule Issues permission can populate the **Due** date field.

Searching by due date

You can use either [basic search](#) or [advanced search](#) to search for issues by their Due Date.

Using simple search

You can search for issues using the search form in Issue Navigator (see [Searching for Issues](#)). There are two ways to search for issues based on the **Due** date field. The first way is using fixed date values, the second is using periods that are relative to the current date.

Fixed date searches

There are two text fields in the search form that allow searching based on the **Due** date field.

- To search for all issues that are due after a certain date, enter the date in the Due After text field. For example, to find all issues that are due after 1st June 2010, enter 1-6-2010 in the Due After field. You can also use the Calendar popup to select a date by clicking the calendar icon to the right of the field.
- To search for issues that are due before a certain date, enter the date in the Due Before text field. For example, to find all issues that are due before 1st July 2010, enter 1-7-2010 in the Due Before field.

To search for issues that are due between two dates, populate both the Due After and the Due Before fields.

Relative period search

It is possible to perform a search that is relative to the time when it is run. For example, it is possible to do a search for issues that are due seven days from now. To do this, enter 7d in the Due Date To text field of the Issue Navigator. If the search is saved and run the next day, the issues that are due in seven days from the time that the search is run will be retrieved. Thus, this search will find all issues that are due within a week every time it is run.

The values that are entered in the Due Date From and Due Date To fields have to conform to a special syntax (described below). However, it is also possible to use the Due Date popup by clicking the icon to the right of the Due Date To text field to specify the search period.

Due Date Popup

Use the Due Date popup to do the following:

- To search for issues that are overdue at the time of the search, select the first radio button, and click **OK**.
- To search for issues that are overdue by more than a certain number of days, populate the text field in the second row, and click **OK**.

- To search for issues that are due in the next certain amount of days, and are not overdue at the time of the search, populate the text field in the third row with the number of days, and choose **and not** from the select box in the third row. Select the third radio button, and click **OK**.
- To search for issues that are due in the next certain amount of days, and are overdue at the time of the search, populate the text field in the third row with the number of days, and choose **and** from the select box in the third row. Select the third radio button, and click **OK**.
- The fourth row of the popup is used for arbitrary period searches. Use the **to** text field to specify the upper bound of the search, and the **from** text field to specify the lower bound of the search. A blank text field means no bound. Populating the text fields in the fourth row actually has the same effect as populating the Due Date From and Due Date To text boxes. The syntax is described below.

Relative Period Search Syntax

The Due Date From and Due Date To fields use a special syntax to denote time period bounds. The syntax uses numbers and abbreviations that follow the numbers to represent what the numbers actually mean. The abbreviations are "w" for weeks, "d" for days, "h" for hours, and "m" for minutes. For example, to specify 10 days in the future, use "10d" or "1w and 3d". To specify a period bound in the past, prefix the value with the "-" sign. For example, to specify 2 days, 4 hours, and 3 minutes ago, use "-2d 4h 3m".

Using advanced search

You can also use Jira Query Language (JQL) to search for issues by due date see [Advanced Searching](#), and particularly the documentation on the `Due` field.

Logging work on issues

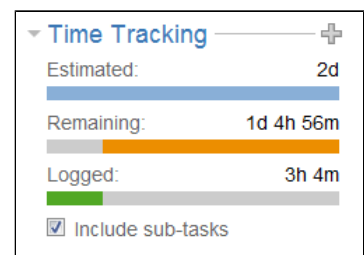
In Jira Service Management, you use Service Level Agreements (SLAs) configured by your administrator to help you track how well you're meeting customer expectations (e.g. responding to a request within 4 hours). You can use the Time Tracking feature in addition to SLAs to generate a workload report when you're working on a customer request with other agents, or when you need to track time spent fixing a problem that affects multiple customer requests.

On this page:

- [Before you begin](#)
- [Setting a time estimate for an issue](#)
- [Logging work on an issue](#)
- [Editing a work log entry](#)
- [Deleting a work log entry](#)
- [Customized Jira installations](#)

Here's how time tracking appears on an issue:

- The Estimated field displays the amount of time originally anticipated to resolve the issue
- The Remaining field displays the amount of time currently anticipated to resolve the issue
- The Logged field displays the amount of time logged working on the issue so far
- Choosing to include sub-tasks displays the aggregated time of an issue and all its sub-tasks



When you log time for the first time, the time spent is subtracted from the original estimate, and the resulting value is automatically presented in the remaining estimate. When subsequent work is logged, any time spent is subtracted from the remaining estimate.

Before you begin

- Make sure your Jira administrator has enabled the [Time Tracking](#) feature.
- Make sure you have the Work on Issues, Delete Work Logs, and Edit Work Logs project permissions.

i Note that anyone with the Browse Project permission can view time tracking information on an issue.


Setting a time estimate for an issue

Teams can set a time estimate for an issue in order to calculate how long it will take to solve the issue.

1. Open the issue and select **Edit**.
2. Scroll down the Edit issue window to fill in the following time tracking fields:

Field	Description
Original Estimate	Amount of time you believe is required to solve the issue. If you want to change original estimate values once they have logged work time, ask your Jira administrator to disable legacy mode on time tracking.
Remaining Estimate	Amount of time you believe is required to solve the issue in its current state.

If the Jira time tracking feature is in legacy mode, you will only see the original estimate field if work has not been logged. Once work time has been logged, you will only see the remaining estimate field.

 **Tips:**

- You can specify additional time units after a time value 'X', such as Xw, Xd, Xh, or Xm, to represent weeks (w), days (d), hours (h), and minutes (m), respectively. If you type a number without specifying a time unit (e.g. if you type '2' instead of '2h'), the default time unit that your Jira administrator specified will apply.
- Default conversion rates are 1w = 5d and 1d = 8h.

3. Select **Update**.

When work is first logged against the issue, the **Time Spent** is subtracted from the **Original Estimate**, and the resulting value is automatically presented in the **Remaining Estimate**. When subsequent work is logged, any **Time Spent** is subtracted from the **Remaining Estimate**.

Additionally, once work has been logged on an issue, various reports based on the time tracking information become available.

Logging work on an issue

Once you have started to work on a specific issue, you can log your work by following these steps:

1. Select the issue you want to log time on.
2. Go to **More > Log Work**.
3. Fill in the following **Log Work** fields, and select **Log**:

Log Work field	Description
Time spent	The amount of time spent on the issue. This is the aggregate amount of time that has been logged against this issue.
Date started	Date and time when you started this unit of work.
Remaining estimate	<p>Amount of time anticipated to resolve the issue after completing this unit of work. You can adjust this value using the following options:</p> <ul style="list-style-type: none"> • Adjust Automatically - Adjust the remaining estimate value by subtracting the amount of work logged in the Time Spent field from the remaining estimate current value. • Leave Estimate unset - This option is displayed only if no time estimate has been specified on the issue. You can use this option when you want to keep track of work, but you don't necessarily have a time estimate for an issue. • Use Existing Estimate of - Select this option if you do not want to change the issue remaining estimate value. • Set to - You can adjust the remaining estimate value to the amount of time you specify in this field. • Reduce by - Select this option to manually adjust the remaining estimate value by subtracting the amount of time you specify in this field.

Work description	<p>Type a description related to the achieved work.</p> <p>Comments are copied to the Workflow Description by default, but your Jira administrator can change this option in the 'Copy Comment to Workflow Descriptions' settings. If this setting is disabled:</p> <ul style="list-style-type: none"> • The work log entry may be visible to anyone. If this is a concern, you need to edit this work log entry after creating it to modify its visibility. • You have to manually copy comments to a workflow description once you have logged work.
------------------	--

You can also log work while resolving or closing an issue by closing it and editing the log work fields. Select the padlock icon to set the work logged to be viewable only by members of a particular project role or group.

Editing a work log entry

You can edit your own work log entries if you have been granted the Edit Own Work Logs permission. You can also edit other people's work log entries if you have been granted the Edit All Work Logs permission.

Deleting a work log entry

You can delete your own work log entries if you have been granted the Delete Own Work Logs permission. You can also delete other people's work log entries if you have been granted the Delete All Work Logs permission.

1. Go to the desired issue, and open the **Work Log** tab.
2. Hover over the work log entry to display the actions for the entry on the right side.
3. Select the entry you want to delete, and click the trash can icon. You will be prompted to choose how the Remaining Estimate is affected by deleting the work log:

Option field	Description
Auto adjust	Choose this option to automatically add the time spent value to the current remaining estimate value.
Leave existing estimate	Select this option if you do not want to change the issue remaining estimate value.
Set estimated time remaining	Choose this option to manually set the issue's remaining estimate value to the specified amount.
Increase estimated time remaining	Select this option to increase the estimated remaining.

4. Click **Delete**.

Customized Jira installations

Jira applications can be customized by your Jira administrator by adding the Log Work and Time Tracking fields to the customized screens. This way, you can log work and specify time estimates on the same Jira screen when performing any Jira operation, such as editing, creating an issue, or transitioning an issue to another status.

If you want to work and/or specify time estimates on the same Jira screen:

1. Navigate to the issue and view its details.
2. Perform the customized Jira operation that allows you to log work and specify time estimates on the same Jira screen. For example, assuming that your Jira administrator has added the **Time Tracking** fields to the **Resolve Issue Screen**, and assuming this screen also retains the default **Log Work** fields, select **Workflow > Resolve Issue** at the top of the issue.




- If your Jira administrator has configured the Log Work fields as optional, then you can choose whether or not to log work by checking the Log Work checkbox.
- If your Jira administrator has made logging work mandatory, you will not see the Log Work checkbox, and will instead need to log work when transitioning an issue.

Approving a service project request

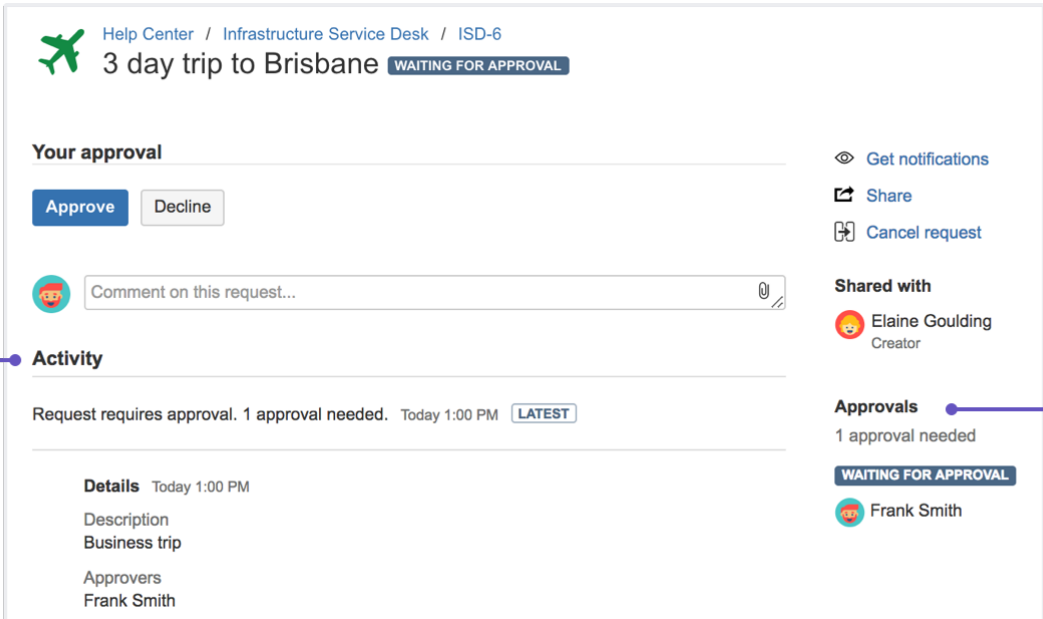
Jira Service Management projects have an option to include an approval step and assign approvers to their issues. You may be asked to approve a service desk request if you've been assigned the role of "approver". When this happens, you'll receive an email notifying you that your approval is required. If the **Request Details** and **Approval buttons** variables have been added to the project's approval notifications template, you can action the request from within the email. If not, there will be a link to the customer portal where you can view and action the request.

How to approve and decline from the customer portal

 **Pro tip:** Ask your Project Administrator to add **Request details** and **Approval buttons** to your approval notifications template, so you can quickly action a request from within your email. See [Managing service project notifications](#).

It's likely that you've upgraded from a version of Jira Service Management prior to 3.12, and the approval notifications template hasn't been updated.

1. Navigate to the service desk customer portal by either selecting the link in your email, or entering the URL.
2. View the approval request and review the supporting information.
3. Leave a comment by selecting **Add** below the comment field. You don't have to leave a comment, but if you're declining a request it helps to let the customer know why you declined it.
4. Select **Approve** or **Decline**.
5. The customer will receive a notification of the selected action you have taken.



The screenshot shows a customer portal interface for a request titled "3 day trip to Brisbane" with a status of "WAITING FOR APPROVAL". The interface includes a "Your approval" section with "Approve" and "Decline" buttons, a comment field, and an "Activity" section. The "Activity" section shows a message: "Request requires approval. 1 approval needed. Today 1:00 PM LATEST". The "Approvals" section shows "1 approval needed" and a list of approvers, including Frank Smith. The "Shared with" section shows Elaine Goulding as the creator. A blue circle with the number "2" points to the "Activity" section, and a blue circle with the number "1" points to the "Approvals" section.

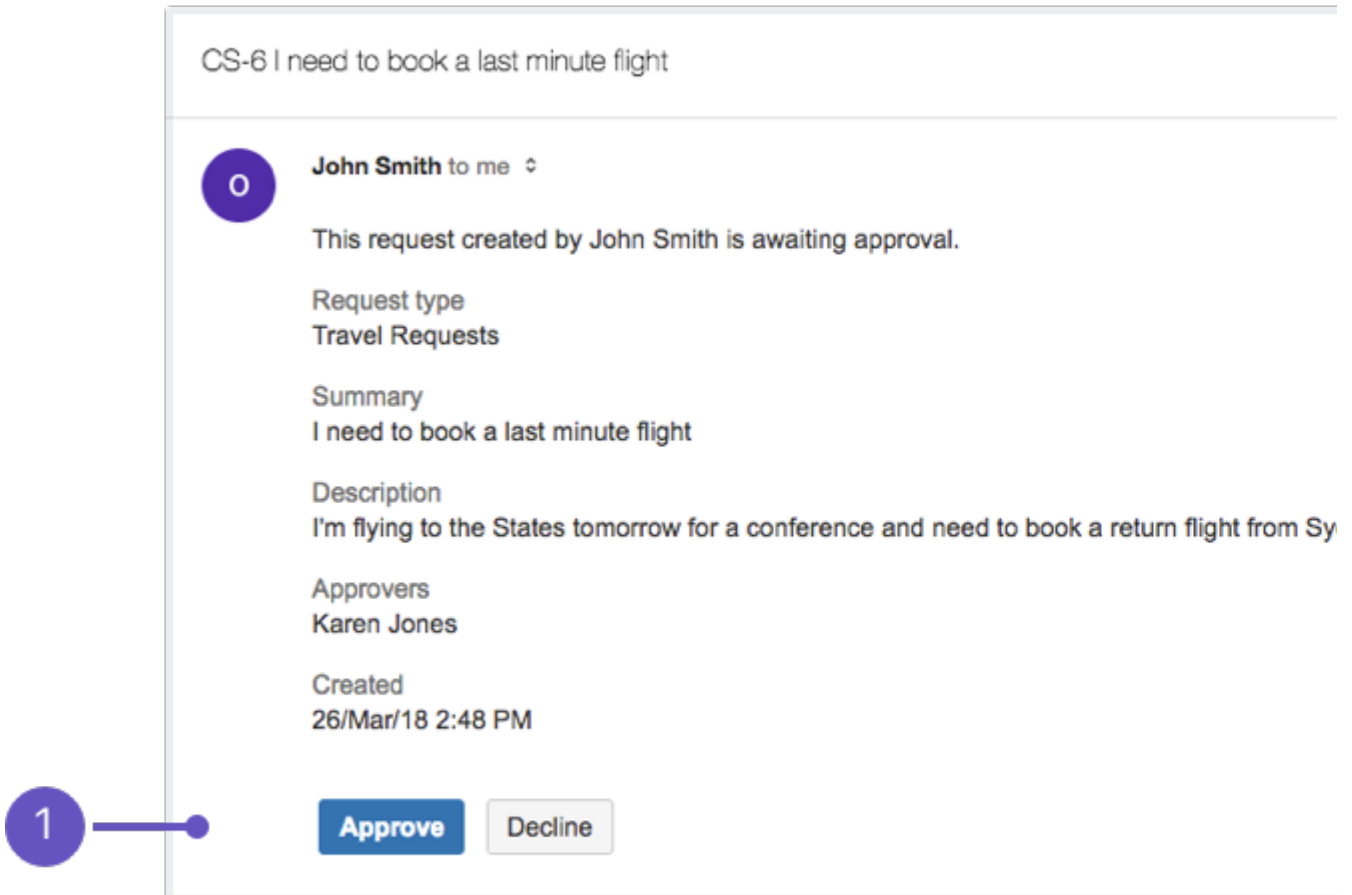
1. **Approvals section**, which lists the appointed approvers for the request. This section only appears when there are pending approvals for the request.
2. When a request is approved, the **Approvals** section disappears, and details about the approval will be added to the **Activity** section.
For example, Your request was **APPROVED** and the status changed to **Waiting for support**.

How to approve and decline from email

Good news! You're using the new default approval notifications template.

1. View the approval request and decide what action you'd like to take.

2. Select either the **Approve** or **Decline** button within the email. If you aren't logged into the customer portal, you will be prompted to do so. You'll then see:
 - a. If you hit **Approve**, a confirmation screen.
 - b. If you hit **Decline**, a screen prompting you to leave a comment and confirm your decision. You don't have to leave a comment, but if you're declining a request it helps to let the customer know why you declined it.
3. The customer will receive a notification of the selected action you have taken.



1. **Approval buttons**, that let the approver action requests from within their email.
2. **Request details**, that show the full details of a request (incl request type, summary, creation date, and the same fields chosen in the request type settings).

If you're the only approver required on the request, and you approve it, the request will be moved to the status defined in the workflow for the approve transition. If there is more than one approval required, the status will remain the same until all approvers have responded, and your approval will be noted on the request.

If you decline a request (or any of the approvers decline it), it's automatically moved to the status as defined in the workflow for the decline transition, and your response is noted on the request.

Customizing the issues in a project

Issues are the packets of work that need to be completed in a project. These issues are made up of issue fields, and the issue fields contain data about the issue. This data is important, as it helps define the issue, and can contain important information about the issue, such as a summary, a description, due dates, and when and where the work is required. This information is presented on a screen. A screen groups all available fields (or a subset of all available fields) defined in Jira applications, and organizes them for presentation to a user. Jira Service Management allows you to customize the configuration and behavior of issues to better suit the needs of your customers and agents. You may choose to:

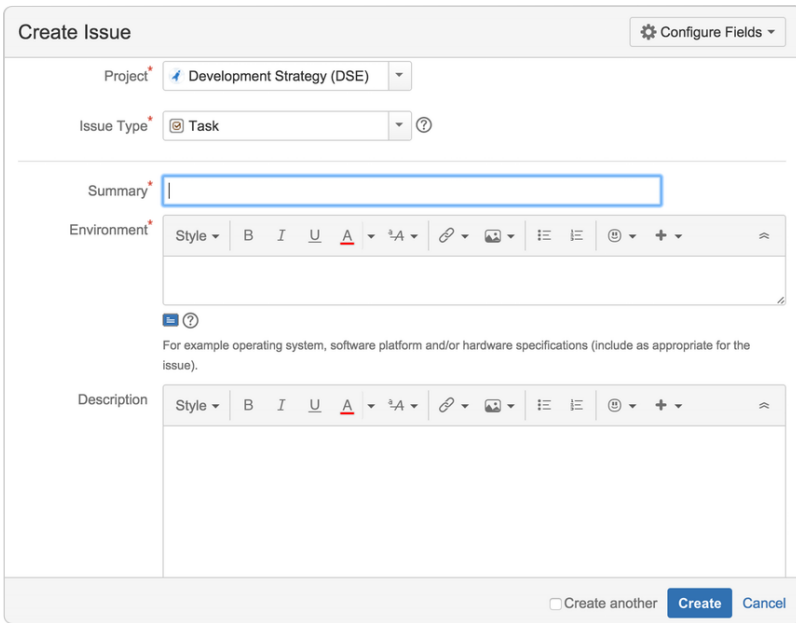
- Change a field's behavior (such as change a field's description, make a field hidden or visible, or make a field required or optional)
- Add your own values for fields that have default values assigned (e.g. Resolution and Status)
- Create new 'custom' fields
- Configure different renderers for (some) fields
- Change the position fields on a screen
- Choose which screen should be displayed for each issue operation (e.g. 'Create Issue', 'Edit Issue') or workflow transition (e.g. Resolve Issue, Close Issue)

A simple example of how customizing an issue could benefit your team could be marking fields as 'Required' when an issue is created. This would ensure you always capture the required information you need to get the work done to resolve the issue. If you couple this with positioning the required fields at the top of the screen, and even hiding fields you know the issue creator won't use, you'll make sure your users can see and complete the required fields as quickly as possible.

You can make this...

The screenshot shows the 'Create Issue' form in Jira Service Management. The form is titled 'Create Issue' and has a 'Configure Fields' button in the top right. It shows fields for Project (Development Strategy (DSE)), Issue Type (Task), Summary (required), Account (Please select), Due Date, Priority (Medium), and Environment (with a rich text editor). The Environment field has a tooltip that reads: 'For example operating system, software platform and/or hardware specifications (include as appropriate for the issue)'. At the bottom, there are buttons for 'Create another', 'Create', and 'Cancel'.

into this...



Jira administrators and project administrators have different permissions when it comes to customizing issues.

Project administrators

As a project administrator, you're able to customize some aspects of the issues in your project if:

- you have the *Extended project administration* permission, which is enabled by default (you can check that in **Project settings > Permissions**),
- the screens you make the changes on must not be used by other projects, or used as a transition screen in a workflow.
- the screen isn't the default Jira screen (no-one can edit these screens).


If a screen is shared by another project, you'll see this information when you view the screen. If your screen suits the criteria above, you can:

- add and remove tabs that will appear on a screen, and edit the name of the tab,
- add, remove and rearrange system fields, and
- add, remove and rearrange existing custom fields, but you can't create custom fields.

Editing a screen

1. Select **Projects** and choose the project whose screen/s you want to view.
2. Select **Project settings** in the sidebar.
3. Select **Screens** in the Project settings sidebar.

You'll see the screen scheme/s used by your project, and the issue types that use that scheme. Expand the screen scheme to see the screens associated with that scheme and issue types. Select the screen you wish to edit. You can perform the following actions:

Action	Instructions
Add a tab	Click Add Tab  . Enter the name of the new tab in the dialog that appears and click Add .
Move a tab	Hover over the dotted part of the tab (next to the tab name) and drag the tab to the desired position.
Rename a tab	1. Hover over the tab name and click the pencil icon . 2. Enter the new name and click OK .

Delete a tab	Hover over the tab name and click the X .
Add a field	<ol style="list-style-type: none">1. Click the tab that you want to add the field to.2. Type the name of the field in the drop-down displayed at the bottom of the current fields. Field suggestions will appear as you type.3. Click Add Field to add it to the current tab.
Move a field	Hover over the dotted part of the field (next to the field name) and drag the field to the desired position. Move a field to a different tab by dragging it to the name of the tab and dropping it.
Remove a field	Hover over the field and click the X that appears.

Jira administrators

As a Jira administrator, you can view more conceptual information on [customizing issues](#) in the Jira administrator's documentation.

Searching for issues


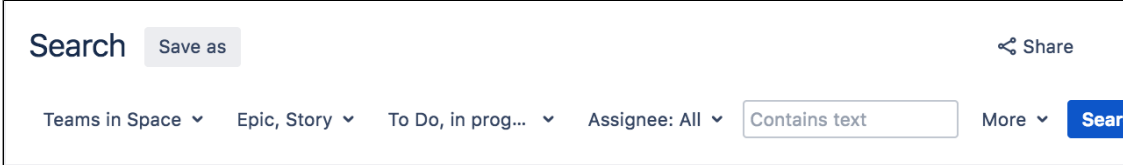

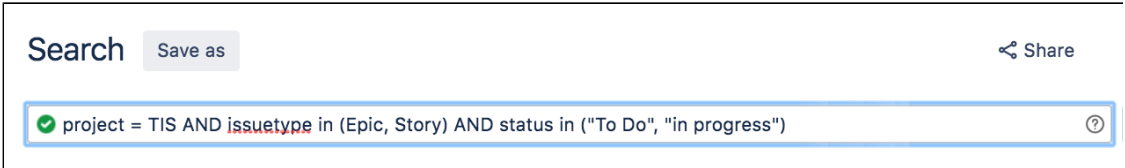
Can't find the customer issue you've been working on? This page will show you how to search for issues in Jira Service Management. Any agent can search for issues, although they will only see results from projects they have access to. You'll find a step-by-step guide below that will show you how to run a search and use the search results. If you want more details on anything described on this page, see the related topics at the bottom of the page.

On this page:

- [1. Define your search criteria](#)
- [2. Change your view of the search results](#)
- [3. Working with the search results](#)
- [4. Save your search](#)
- [Good to know](#)
- [Next steps](#)

1. Define your search criteria

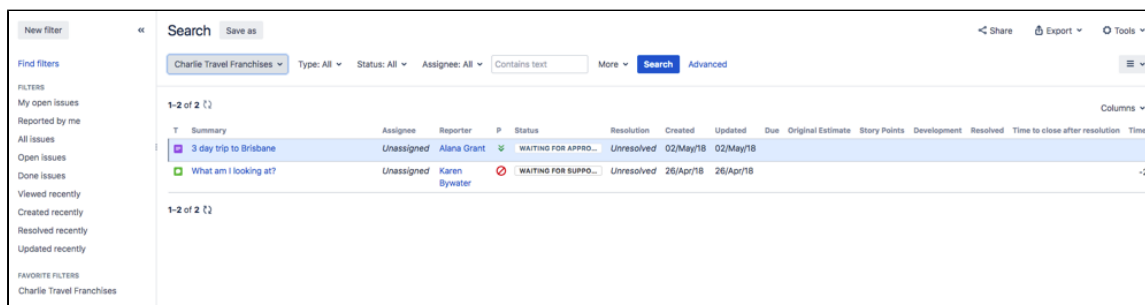
The first step in searching for issues is to define the criteria for your new search. You can define your search criteria in three different ways: using the quick search, using the basic search, or using the advanced search.

Quick search	<p>The quick search is the fastest way to define search criteria. However, it is less precise than other complex queries (e.g. <code>project = Jira AND status = Open AND priority = High</code>). If your search criteria is not complex, for example, you know the project key and some key words.</p> <p>To use the quick search: Enter your search criteria in the search box in the header bar of Jira</p> <p><i>Tip: If you know the issue key or project key, enter it before other search terms, e.g. "JIRA help"</i></p>
Basic search	<p>The basic search is more precise than the quick search, but easier to use than the advanced search. It provides a user-friendly interface that lets you define complex queries, without needing to know how to use JQL (searching).</p> <p>To use the basic search: Navigate to Issues (in header) > Search for issues, then enter your search criteria</p> <p><i>Tip: If the advanced search is shown instead of the basic search, click Basic next to the  icon</i></p> <div data-bbox="339 1288 1469 1451"></div>
Advanced search	<p>The advanced search is the most powerful of the three search methods. You can specify criteria that are not defined in the other searches (e.g. <code>ORDER BY</code> clause). However, you need to know how to construct complex queries using the Jira Query Language (JQL) to use this feature.</p> <p>To use the advanced search: Navigate to Issues (in header) > Search for issues, then enter your JQL query</p> <p><i>Tip: If the basic search is shown instead of the advanced search, click Advanced next to the  icon</i></p> <div data-bbox="339 1697 1469 1863"></div>

2. Change your view of the search results

You have crafted the perfect search criteria and run the search. Your search results will be displayed in the issue navigator. The issue navigator allows you to change how the search results are displayed. For example, you may want to bring high priority issues to the top or hide certain fields.

- **Change the sort order:** Click the column name.
- **Show/hide columns:** Click **Columns** and choose the desired columns.



3. Working with the search results

You've got the search results displaying the way that you want. Now you can work with the actual issues in the search results. The issue navigator lets you action individual issues, as well as the entire set of issues returned by your search.

Individual issues:

- **View the issue:** Click the issue key or name.
- **Action individual issues:** Click the cog icon next to the issue row and select an option.

All issues in the search results:

- **Export the search results to different formats, like CSV and XML:** Click **Export** and select the desired format.
- **Share the search results:** Click **Share**, then enter the recipient's details.
- **Create an RSS feed:** Click **Export > RSS (Issues)** or **RSS (Comments)**.
- **Bulk modify issues in search results:** Click **Tools** and select **all <n> issue(s)** under **Bulk Change**.

4. Save your search

If you frequently run the same search, you can save the search criteria as a filter. This saves you from having to manually redefine the search criteria every time. Jira applications also include a number of predefined system filters for common queries, such as 'My Open Issues', 'Reported by Me', 'Recently Viewed', and 'All Issues'.

To save your search as a filter: On the search results page, click **Save as** and enter a name for the filter. Your new filter will be shown in the left panel with your other favorite filters, filters shared with you, and the system filters. To run a filter, just click it.

Good to know

Keep in mind that your search won't include issues that have been archived. These are removed from Jira's index, and can't be searched for like other issues.

Next steps

Read the following related topics:

- [Quick searching](#)
- [Basic searching](#)
- [Advanced searching](#)
- [Saving your search as a filter](#)
- [Working with search results](#)

Basic searching

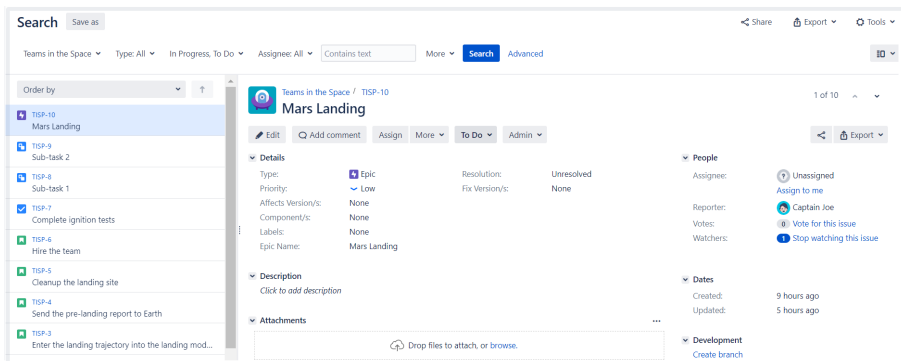
The basic search provides a user-friendly interface that lets you define complex queries, without needing to know how to use JQL (advanced searching).

- If you don't have complex search criteria, you may want to use [quick search](#) instead.
- If you are comfortable with the Jira Query Language (JQL), you may want to use [advanced search](#) instead. This search is more powerful than than the basic search.

On this page:

- [Basic searching](#)
- [Running a saved search](#)
- [Troubleshooting](#)
- [Next steps](#)

Screenshot: Basic search



Basic searching

1. Choose **Issues** > **Search for issues**.

- If there are existing search criteria, click the **New filter** button to reset the search criteria.
- If the advanced search is shown instead of the basic search, click **Basic** (next to the **Search** button).

In general, a query created using basic search will be able to be translated to advanced search, and back again. However, a query created using advanced search may not be able to be translated to basic search, particularly if:

- the query contains an OR operator (note you can have an IN operator and it will be translated, e.g. `project in (A, B)`)
 - i.e. even though this query: `(project = JRA OR project = CONF)` is equivalent to this query: `(project in (JRA, CONF))`, only the second query will be translated.
- the query contains a NOT operator
- the query contains an EMPTY operator
- the query contains any of the comparison operators: `!=`, `IS`, `IS NOT`, `>`, `>=`, `<`, `<=`
- the query specifies a field and value that is related to a project (e.g. version, component, custom fields) and the project is not explicitly included in the query (e.g. `fixVersion = "4.0"`, without the `AND project=JRA`). This is especially tricky with custom fields since they can be configured on a Project/Issue Type basis. The general rule of thumb is that if the query cannot be created in the basic search form, then it will not be able to be translated from advanced search to basic search.

2. Enter the criteria for the search. You can search against specific fields and/or search for specific text.

- If you are searching against a field and can't find the field you want, or the field is displaying greyed out text, see the [Troubleshooting section](#) below.
- If you are searching for text, you can use special characters and modifiers in your search text, such as wildcards and logical operators. See [Search syntax for text fields](#).

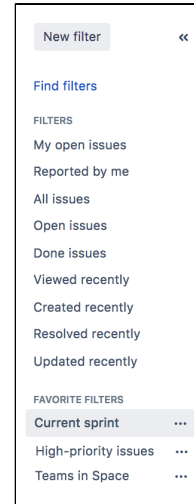
3. The search results will automatically update in the issue navigator, unless your administrator has disabled automatic updates of search results. If so, you will need to click the **Update** button on the field drop-down after every change.

Running a saved search

Saved searches (also known as [filters](#)) are shown in the left panel, when using basic search. If the left panel is not showing, hover your mouse over the left side of the screen to display it.

To run a filter, e.g. **My Open Issues**, simply click it. The search criteria for the basic search will be set, and the search results will be displayed.

Note, clicking the **Recently Viewed** filter will switch you to the advanced search, as the basic search cannot represent the `ORDER BY` clause in this filter.



Troubleshooting

Some fields are only valid for a particular *project/issue type context*. For these fields, you must select the applicable project/issue type. Otherwise, the field is not available for selection.

Some fields are only valid for a particular *project/issue type context*. If you choose a field in your search, then remove all projects/issue types that reference the field, then the field is invalid. The invalid field does not apply to your search and displays in grey text.

Some field values are only valid for a particular *project/issue type context*. For example, you may have configured a project to use a status *In Review* in its workflow. If you select this project and status in your search, then change the search to filter for a project that doesn't use *In Review*, the status will be invalid and ignored in the search.

Your search results will always update automatically whenever any fields are changed, provided that your administrator has not disabled automatic updates of search results. Ask your administrator whether they have disabled automatic updates of search results.

Next steps

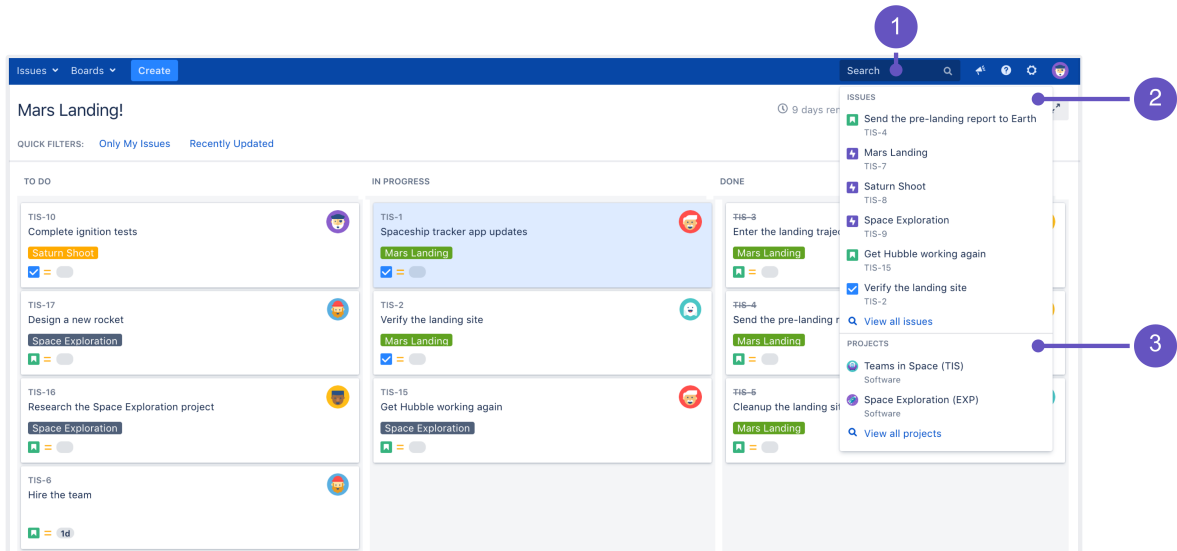
Read the following related topics:

- [Searching for issues](#)
- [Advanced searching](#)
- [Saving your search as a filter](#)
- [Working with search results](#) find out how to use the issue navigator, export your search results, bulk modify issues, and share your search results.

Quick searching

Sometimes, you just want to be able to get to the particular issue that you're interested in. Other times, you can't remember what the issue was, but you remember that it was an open issue, assigned to you, or you have its name on the tip of your tongue. Quick search can help you in these scenarios.

The **Search** box is located at the top right of your screen, in the Jira header bar. To use quick search, just start typing what you're looking for.



1. **Search:** Click anywhere in the box to display your recent work, or start typing to search through all your issues and projects.
2. **Issues:** Recent issues (before searching), or issues that match your search.
3. **Projects:** Recent projects (before searching), or projects that match your search.

✔ Using quick search by many users at once can affect performance. You can limit the number of concurrent searches, or monitor how your users are searching in real-time. [Learn more](#)

Understanding quick searching

Read the following topics to learn how to get the most out of quick searching:

[Jumping to an issue](#) | [Searching as you type](#) | [Free-text searching](#) | [Smart querying](#)

Jumping to an issue

If you type in the **key** of an issue, you will jump straight to that issue. For example, if you type in 'ABC-107' (or 'abc-107'), and press the **Enter** button, you will be redirected to the issue 'ABC-107'.

In many cases, you do not even need to type in the full key, but just the numerical part. If you are currently working on the 'ABC' project, and you type in '123', you will be redirected to 'ABC-123'.

Searching as you type

When you start typing the word you're looking for, the quick search will react instantly by showing and refreshing the list of most relevant results. To display these results, your search term is matched against the following fields:

- Summary (projects and issues)
- Description (issues)

Free-text searching

You can additionally search through comments or use extra operators for fuzzy or wildcard search. These results won't be displayed as 'instant results', but you can view them after pressing **Enter** in the search box.

You can combine free-text and keywords together, e.g. "my closed test tasks". You can also use wildcards, e.g. "win*8".

For more information on free-text searching, see [Search syntax for text fields](#).

Smart querying

Quick search also enables you to perform 'smart' searches with minimal typing. For example, to find all the open bugs in the 'TEST' project, you could simply type 'test open bugs' and quick search would locate them all for you.

Your search results will be displayed in the Issue Navigator, where you can view them in a variety of useful formats (Excel, XML, etc).

The search terms that quick search recognizes are:

Search Term	Description	Examples
my	Find issues assigned to me.	my open bugs
r:	Find issues reported by you, another user or with no reporter, using the prefix <i>r:</i> followed by a specific reporter term, such as <i>me</i> , a username or <i>none</i> . <i>Note that there can be no spaces between "r:" and the specific reporter term.</i>	r:me finds issues reported by you. r:samuel finds issues reported by the user whose username is "samuel". r:none finds issues with no reporter.
<project name> or <project key>	Find issues in a particular project.	test project TST tst
overdue	Find issues that were due before today.	overdue
created: updated: due:	Find issues with a particular Created, Updated, or Due Date using the prefixes <i>created:</i> , <i>updated:</i> , or <i>due:</i> , respectively. For the date range, you can use <i>today</i> , <i>tomorrow</i> , <i>yesterday</i> , a single date range (e.g. '-1w'), or two date ranges (e.g. '-1w,1w'). Note that date ranges cannot have spaces in them. Valid date/time abbreviations are: 'w' (week), 'd' (day), 'h' (hour), 'm' (minute).	created:today created:yesterday updated:-1w finds issues updated in the last week. due:1w finds issues due in the next week. due:-1d,1w finds issues due from yesterday to next week. created:-1w,-30m finds issues created from one week ago, to 30 minutes ago. created:-1d updated:-4h finds issues created in the last day, updated in the last 4 hours.
<priority>	Find issues with a particular Priority.	blocker major trivial

<issue type>	Find issues with a particular Issue Type. Note that you can also use plurals.	bug task bugs tasks
<resolution>	Find issues with a particular Resolution.	fixed duplicate cannot reproduce
c:	Find issues with a particular Component(s). You can search across multiple components. <i>Note that there can be no spaces between "c:" and the component name.</i>	c:security finds issues with a component whose name contains the word "security".
v:	Find issues with a particular Affects Version(s). To find all issues belonging to a 'major' version, use the wildcard symbol '*'. <i>Note that there can be no spaces between "v:" and the version name.</i>	v: 3.0 finds issues that match the following versions (for example): <ul style="list-style-type: none"> • 3.0 • 3.0 eap • 3.0 beta <p>...but will not match against the following versions (for example):</p> <ul style="list-style-type: none"> • 3.0.1 • 3.0.0.4 <p>That is, it will match against any version that contains the string you specify followed immediately by a space, but not against versions that do not contain a space immediately after the string you specify.</p>
ff:	Find issues with a particular Fix For Version(s). Same usage as v: (above).	
*	Wildcard symbol '*'. Can be used with v: and ff:.	v: 3.2* finds any issue whose version number is (for example): <ul style="list-style-type: none"> • 3.2 • 3.2-beta • 3.2.1 • 3.2.x

i In Mozilla-based browsers, try creating a bookmark with URL `http://<your-Jira-site>/secure/QuickSearch.jsps?searchString=%s` (substituting <your-Jira-site> with your Jira instance's URL) and keyword (such as 'j'). Now, typing 'j my open bugs' in the browser URL bar will search your Jira instance for your open bugs. Or simply type your search term in the Quick Search box, then right-click on the Quick Search box (with your search term shown) and select "Add a Keyword for this search...".

Searching issues from your browser's search box

If you are using Firefox or Internet Explorer 8 (or later), you can add your Jira instance as a search engine/provider via the drop-down menu next to the browser's search box. Once you add your Jira instance as a search engine/provider in your browser, you can use it at any time to conduct a Quick Search for issues in that Jira instance.

OpenSearch

Jira supports this browser search feature as part of the autodiscovery part of the [OpenSearch](#) standard, by supplying an [OpenSearch description document](#). This is an XML file that describes the web interface provided by Jira's search function. Any [client applications](#) that support OpenSearch will be able to add Jira to their list of search engines.

Next steps

Read the following related topics:

- [Searching for issues](#)

Advanced searching

The advanced search allows you to build structured queries using the Jira Query Language (JQL) to search for issues. You can specify criteria that cannot be defined in the quick or basic searches. For example, you can use the `ORDER BY` clause to sort Jira issues either in descending or ascending order or narrow down your search results for the desired date range.


On this page:

- [Using advanced search](#)
- [Reference](#)
- [Running a saved search](#)

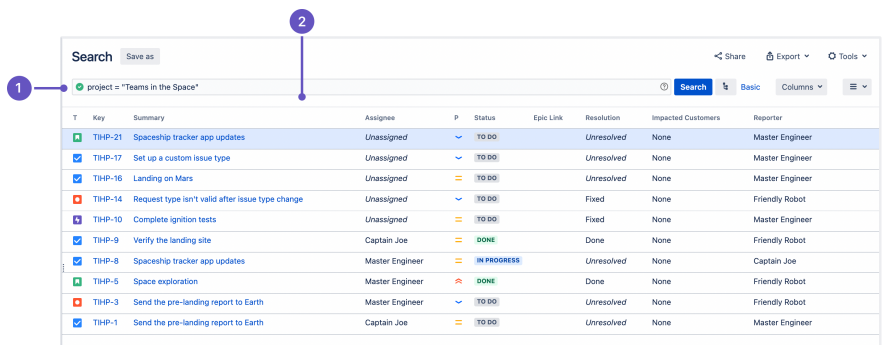
 Learn more about searching in Jira from [JQL: The most flexible way to search Jira \(on the Atlassian blog\)](#)

Before using the advanced search, consider the following:

- If you don't have complex search criteria, you may want to use [quick search](#) instead.
- If you are not comfortable with the Jira Query Language (JQL), you may want to use [basic search](#) instead.

 Note that JQL is not a database query language, even though it uses SQL-like syntax.

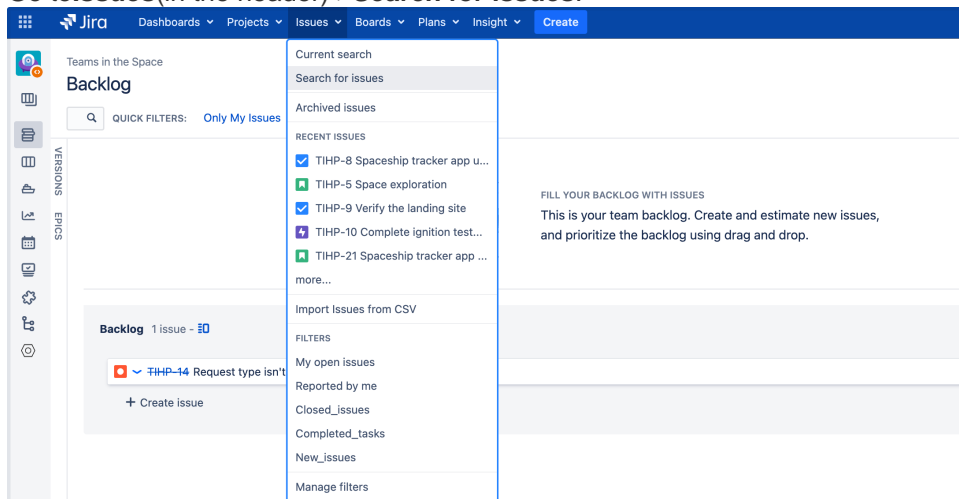
The following is the example of an advanced search query in Jira that returns all issues for the **Teams in the Space** project.



1. **JQL query** that refines the search results.
2. A list of **Jira issues** that match the search criteria.

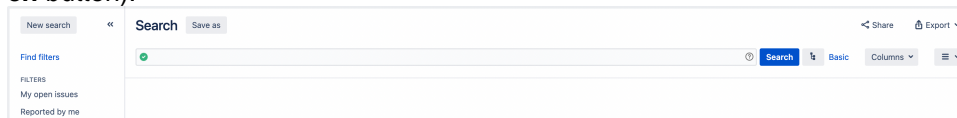
Using advanced search

1. Go to **Issues** (in the header) > **Search for issues**.



- If there are existing search criteria, select the **New search** button to reset the search criteria.

- If the basic search is shown instead of the advanced search, select **Advanced**(next to the **Search** button).



i If you cannot switch to an advanced search, check out the [following](#) section.

2. Enter your JQL query. As you type, Jira will offer a list of "auto-complete" suggestions based on the context of your query. Note that, auto-complete suggestions only include the first 15 matches, displayed alphabetically, so you may need to enter more text if you can't find a match.

If you don't see auto-complete suggestions, this might happen because of the following:

- Your administrator may have disabled the "JQL Auto-complete" feature for your Jira instance.
- Auto-complete suggestions are not offered for function parameters.
- Auto-complete suggestions are not offered for all fields. Check the [fields](#) reference to see which fields support auto-complete.

3. Press Enter or select **Search** to run your query. Your search results will display in the issue navigator.

Switch between basic and advanced search

In general, a query created using basic search will be able to be translated to advanced search, and back again. However, sometimes a query created using an advanced search may not be able to be translated into a basic search. Expand the following section for details.

You might not switch between two searches if:

- the query contains an OR operator.

i You can have an IN operator and it will be translated, e.g. `project in (A, B)`. Even though this query: `(project = JRA OR project = CONF)` is equivalent to this query: `project in (JRA, CONF)`, only the second query will be translated.

- the query contains a NOT operator.
- the query contains an EMPTY operator.
- the query contains any of the comparison operators: `!=`, `IS`, `IS NOT`, `>`, `>=`, `<`, `<=`.
- the query specifies a field and value that is related to a project (e.g. version, component, custom fields) and the project is not explicitly included in the query (e.g. `fixVersion = "4.0"`, without the `AND project=JRA`). This is especially tricky with custom fields since they can be configured on a Project/Issue Type basis. The general rule of thumb is that if the query cannot be created in the basic search form, then it won't be able to be translated from advanced search to basic search.

Understanding advanced searching

Read the following topics to learn how to get the most out of advanced searching:

- [Constructing JQL queries](#)
- [Precedence in JQL queries](#)
- [Restricted words and characters](#)
- [Performing text searches](#)

Constructing JQL queries

A simple query in JQL (also known as a "clause") consists of a *field*, followed by an *operator*, followed by one or more *values* or *functions*.

Example 1

This query will find all issues in the TEST project.

```
project = "TEST"
```

This query will find all issues in the TEST project. It uses the `project` field, the `EQUALS` operator, and the value TEST.

Example 2

A more complex query might look like this:

```
project = "TEST" AND assignee = currentUser()
```

This query will find all issues in the TEST project where the assignee is the currently logged in user. It uses the `project` field, the `EQUALS` operator, the value TEST, the `AND` keyword and the `currentUser()` function.

Example 3

A JQL query that will search for more than one value of a specific field. This query will find all issues of type Bug, which have accessibility and "3rd-party apps" values for the Component field:

```
issuetype = Bug AND component in (accessibility, "3rd-party apps")
```

The query uses the `issuetype` field, the `EQUALS` operator, the value Bug, the `AND` keyword, the `component` field, and the `IN` operator.

Example 4

A JQL query that will find issues created since the start of the current year and updated since the start of the current month:

```
project = "Analytics" and created > startOfYear() and updated > startOfMonth()
```

Example 5

A JQL query that will find any issues that are created in the Test project and contain the "pre-landing report" text in a summary or description:

```
project = "Test" AND text ~ "pre-landing report"
```

For more information on fields, operators, keywords and functions, see the [Reference section](#).

Precedence in JQL queries

Precedence in JQL queries depends on keywords that you use to connect your clauses. For example, a clause can be: `project = Teams in Space`. The easiest way to look at this is to treat the `AND` keyword as the one grouping clauses, and `OR` as the one separating them. The `AND` keyword takes precedence over other keywords, because it groups clauses together, essentially turning them into one combined clause.

Example 1

```
status=resolved AND project=Teams in Space OR assignee=captainjoe
```

This query will return all resolved issues from the Teams in Space project (clauses grouped by AND), and also all existing issues assigned to captainjoe. The clause after the OR keyword is treated as separate.

Example 2

```
status=resolved OR project="Teams in Space" AND assignee=captainjoe
```

This query, on the other hand, will return captainjoes issues from the Teams in Space project (clauses grouped by AND), and also all existing resolved issues (a clause separated by OR).

Example 3

```
status=resolved OR projects="Teams in Space" OR assignee=captainjoe
```

When you only use the OR keyword, all clauses will be treated as separate, and equal in terms of precedence.

Setting the precedence

You can set precedence in your JQL queries by using parentheses. Parentheses will group certain clauses together and enforce precedence.

Example 1

As you can see in this example, parentheses can turn our example JQL query around. This query would return resolved issues that either belong to the Teams in Space project or are assigned to captainjoe.

```
status=resolved AND (project="Teams in Space" OR assignee=captainjoe)
```

Example 2

If you used parentheses like in the following example, they wouldnt have any effect, because the clauses enclosed in parentheses were already connected by AND. This query would return the same results with or without the parentheses.

```
(status=resolved AND project="Teams in Space") OR assignee=captainjoe
```

Restricted words and characters

Reserved characters

JQL has a list of reserved characters:

space(" ")	+	.	,	;	?		*	/	%	^	\$	#	@	[]
------------	---	---	---	---	---	--	---	---	---	---	----	---	---	---	---

If you wish to use these characters in queries, you need to:

- Surround them with quote-marks. You can use either single quotation marks (') or double quotation marks (").
and
- If you are searching a **text field** and the character is on the list of [special characters in text searches](#), precede them with two backslashes. This will let you run the query that contains a reserved character, but the character itself will be ignored in your query. For more information, see **Special characters in Search syntax for text fields**.

For example:

- version = "[example]"

- `summary ~ "\\[example\\]"`

Reserved words

JQL also has a list of reserved words. These words need to be surrounded by quotation marks (single or double) if you wish to use them in queries.

"abort", "access", "add", "after", "alias", "all", "alter", "and", "any", "as", "asc", "audit", "avg", "before", "begin", "between", "boolean", "break", "by", "byte", "catch", "cf", "char", "character", "check", "checkpoint", "collate", "collation", "column", "commit", "connect", "continue", "count", "create", "current", "date", "decimal", "declare", "decrement", "default", "defaults", "define", "delete", "delimiter", "desc", "difference", "distinct", "divide", "do", "double", "drop", "else", "empty", "encoding", "end", "equals", "escape", "exclusive", "exec", "execute", "exists", "explain", "false", "fetch", "file", "field", "first", "float", "for", "from", "function", "go", "goto", "grant", "greater", "group", "having", "identified", "if", "immediate", "in", "increment", "index", "initial", "inner", "inout", "input", "insert", "int", "integer", "intersect", "intersection", "into", "is", "isempty", "isnull", "join", "last", "left", "less", "like", "limit", "lock", "long", "max", "min", "minus", "mode", "modify", "modulo", "more", "multiply", "next", "noaudit", "not", "notin", "nowait", "null", "number", "object", "of", "on", "option", "or", "order", "outer", "output", "power", "previous", "prior", "privileges", "public", "raise", "raw", "remainder", "rename", "resource", "return", "returns", "revoke", "right", "row", "rowid", "rownum", "rows", "select", "session", "set", "share", "size", "sqrt", "start", "strict", "string", "subtract", "sum", "synonym", "table", "then", "to", "trans", "transaction", "trigger", "true", "uid", "union", "unique", "update", "user", "validate", "values", "view", "when", "whenever", "where", "while", "with"

 If you're a Jira admin, note that this list is hard coded in the `JqlStringSupportImpl.java` file.

Performing text searches

You can use Lucene's text-searching features when performing searches on the following fields by using the `CONTAINS` operator.

- Summary.
- Description.
- Environment.
- Comments.
- Custom fields that use the "Free Text Searcher". These are custom fields of the following built-in custom field types: Free Text Field, Text Field, and Read-only Text Field.

When searching for text fields, you can also use single and multiple character [wildcard searches](#). For more information, see [Search syntax for text fields](#).

Differences between day and time search

A day (1d) and time (24h) values are differently calculated in a query and don't return the same results:

- If you specify "1d", the start of the day will start calculating at 00:00 of the server timezone unless you also add the exact time. "1d" will also include the current day if you execute the query now. It doesn't take into account the amount of time relative to the time you had executed the query (24 hours from the time you executed the JQL).
- If you use "24h", it will start calculating from the hour when you executed it (-24 hours from the time you run the JQL).

Example

Let's assume that you updated an issue's status to "Closed" yesterday at 3 PM. You run the following queries at 1 PM today:

- `status changed to "Closed" after -1d` won't return the closed issue. However, it'll return the result if you run `status changed to "Closed" after -2d`.
- `status changed to "Closed" after -24h` will return the closed issue.

Reference

Here you can find a brief overview of Jira fields, operators, keywords, and functions used to compose JQL queries. For detailed description and examples of their usage for advance searching, check the links from the **Reference** column.

	Description	Reference
--	-------------	-----------

<p>Fields</p>	<p>A field in JQL is a word that represents a Jira field (or a custom field that has already been defined in Jira). You can perform an advanced search on your Jira fields to look for issues created on, before, or after a particular date (or date range) and time.</p>	<p>To view a detailed information about fields and how to use them for advanced searching, check out Fields reference page.</p> <ul style="list-style-type: none"> • affectedVersion • approvals • assignee • attachments • category • comment • component • created • creator • customFieldName • "Customer Request Type" • description • due • environment • "epic link" • filter • fixVersion • issueKey • Issue link type • labels • lastViewed • level • originalEstimate • parent • priority • project • remainingEstimate • reporter • request-channel-type • request-last-activity-time • resolution • resolved • sprint • status • summary • text • timeSpent • type • updated • voter • votes • watcher • watchers • worklogAuthor • WorklogComment • WorklogDate • WorkRatio
----------------------	--	--

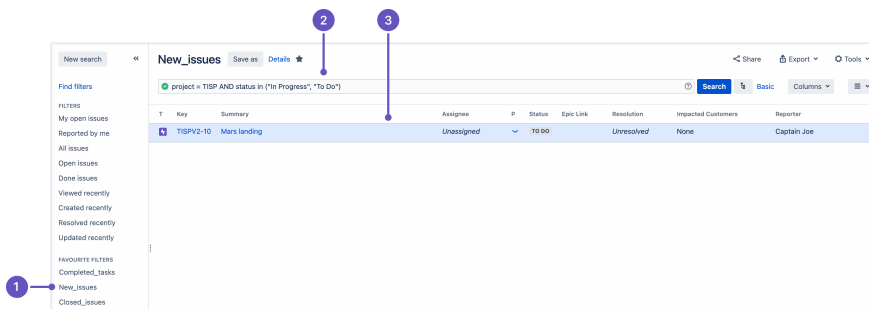
<p>Operators</p>	<p>An operator in JQL is one or more symbols or words that compare the value of a field on its left with one or more values (or functions) on its right, such that only true results are retrieved by the clause. Some operators may use the NOT keyword.</p>	<p>To view a detailed information about operators and how to use them for advanced searching, check out Operators reference page.</p> <ul style="list-style-type: none"> • EQUALS: = • NOT EQUALS: != • GREATER THAN: > • GREATER THAN EQUALS: >= • LESS THAN: < • LESS THAN EQUALS: <= • IN • NOT IN • CONTAINS: ~ • DOES NOT CONTAIN: !~ • IS • IS NOT • WAS • WAS IN • WAS NOT IN • WAS NOT • CHANGED
<p>Keywords</p>	<p>A keyword in JQL is a word or phrase that does (or is) any of the following:</p> <ul style="list-style-type: none"> • joins two or more clauses together to form a complex JQL query. • alters the logic of one or more clauses. • alters the logic of operators. • has an explicit definition in a JQL query. • performs a specific function that alters the results of a JQL query. 	<p>To view a detailed information about keywords and how to use them for advanced searching, check out Keywords reference page.</p> <ul style="list-style-type: none"> • AND • OR • NOT • EMPTY • NULL • ORDER BY

<p>Functions</p>	<p>A function in JQL appears as a word followed by parentheses, which may contain one or more explicit values or Jira fields.</p> <p>A function performs a calculation on either specific Jira data or the function's content in parentheses, such that only true results are retrieved by the function, and then again by the clause in which the function is used.</p>	<p>To view a detailed information about functions and how to use them for advanced searching, check out Functions reference page.</p> <ul style="list-style-type: none"> • <code>approved()</code> • <code>approver()</code> • <code>cascadeOption()</code> • <code>closedSprints()</code> • <code>componentsLeadByUser()</code> • <code>currentLogin()</code> • <code>currentUser()</code> • <code>earliestUnreleasedVersion()</code> • <code>endOfDay()</code> • <code>endOfMonth()</code> • <code>endOfWeek()</code> • <code>endOfYear()</code> • <code>issueHistory()</code> • <code>issuesWithRemoteLinksByGlobalId()</code> • <code>lastLogin()</code> • <code>latestReleasedVersion()</code> • <code>linkedIssues()</code> • <code>membersOf()</code> • <code>myApproval()</code> • <code>myPending()</code> • <code>now()</code> • <code>openSprints()</code> • <code>pending()</code> • <code>pendingBy()</code> • <code>projectsLeadByUser()</code> • <code>projectsWhereUserHasPermission()</code> • <code>projectsWhereUserHasRole()</code> • <code>releasedVersions()</code> • <code>standardIssueTypes()</code> • <code>startOfDay()</code> • <code>startOfMonth()</code> • <code>startOfWeek()</code> • <code>startOfYear()</code> • <code>subtaskIssueTypes()</code> • <code>unreleasedVersions()</code> • <code>votedIssues()</code> • <code>watchedIssues()</code>
-------------------------	--	---

Running a saved search

You can find saved searches (also known as [Saving your search as a filter](#)) in the left-side panel, when using advanced search. If the left panel is not showing, hover your mouse over the left side of the screen to display it.

To run a filter, such as **New_issues**, select the filter name. The JQL for the advanced search will be set, and the search results will be displayed.



1. A **search saved as a filter**, which returns issues based on the criteria specified in a JQL query.
2. **JQL query** that specifies search criteria.
3. **Issues** that match the search criteria.

If you want to delete a saved search, see [Deleting a filter](#).

Advanced searching - fields reference

This page describes information about fields that are used for advanced searching. A field in JQL is a word that represents a Jira field (or a custom field that has already been defined in your Jira applications). In a clause, a field is followed by an [operator](#), which in turn is followed by one or more values (or [functions](#)). The operator compares the value of the field with one or more values or functions on the right, such that only true results are retrieved by the clause. Note: it is not possible to compare two fields in JQL.

Affected version

Search for issues that are assigned to a particular affects version(s). You can search by version name or version ID (i.e. the number that Jira automatically allocates to a version). Note, it is better to search by version ID than by version name. Different projects may have versions with the same name. It is also possible for your Jira administrator to change the name of a version, which could break any saved filters that rely on that name. Version IDs, however, are unique and cannot be changed.

Syntax	<code>affectedVersion</code>
Field Type	VERSION
Auto-complete	Yes
Supported operators	<p><code>= , != , > , >= , < , <= , ~ , !~</code> <code>IS, IS NOT, IN, NOT IN</code></p> <ul style="list-style-type: none"> • The comparison operators (e.g. ">") use the version order that has been set up by your project administrator, not a numeric or alphabetic order. • For this field, the contain operators (~ and !~) find exact matches, and can be used to search through versions with a wildcard.
Unsupported operators	<code>WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED</code>
Supported functions	<p>When used with the IN and NOT IN operators, this field supports:</p> <ul style="list-style-type: none"> • <code>releasedVersions()</code> • <code>latestReleasedVersion()</code> • <code>unreleasedVersions()</code> • <code>earliestUnreleasedVersion()</code>
Examples	<ul style="list-style-type: none"> • Find issues with an AffectedVersion of 3.14: <code>affectedVersion = "3.14"</code> <i>Note that full-stops are reserved characters and need to be surrounded by quote-marks.</i> • Find issues with an AffectedVersion of "Big Ted": <code>affectedVersion = "Big Ted"</code> • Find issues with an AffectedVersion ID of 10350: <code>affectedVersion = 10350</code>

On this page:

- [Affected version](#)
- [Approvals](#)
- [Assignee](#)
- [Attachments](#)
- [Category](#)
- [Comment](#)
- [Component](#)
- [Created](#)
- [Creator](#)
- [Custom field](#)
- [Customer Request Type](#)
- [Description](#)
- [Due](#)
- [Environment](#)
- [Epic link](#)
- [Filter](#)
- [Fix version](#)
- [Issue key](#)
- [Issue link type](#)
- [Labels](#)
- [Last viewed](#)
- [Level](#)
- [Original estimate](#)
- [Parent](#)
- [Priority](#)
- [Project](#)
- [Remaining estimate](#)
- [Reporter](#)
- [Request channel type](#)
- [Request last activity time](#)
- [Resolution](#)
- [Resolved](#)
- [SLA](#)
- [Sprint](#)
- [Status](#)
- [Summary](#)
- [Text](#)
- [Time spent](#)
- [Type](#)
- [Updated](#)
- [Voter](#)
- [Votes](#)
- [Watcher](#)
- [Watchers](#)
- [Work log author](#)
- [Work log comment](#)
- [Work log date](#)
- [Work ratio](#)

[^ top of page](#)

Approvals

Only applicable if Jira Service Management is installed and licensed, and you're using the Approvals functionality.

Search for issues that have been approved or require approval. This can be further refined by user.

Syntax	<code>approvals</code>
Field Type	USER
Auto-complete	No
Supported operators	=
Unsupported operators	~ , != , !~ , > , >= , < , <= IS , IS NOT , IN , NOT IN , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Supported functions	<ul style="list-style-type: none"> • <code>approved()</code> • <code>approver()</code> • <code>myApproval()</code> • <code>myPending()</code> • <code>pending()</code> • <code>pendingBy()</code>
Examples	<ul style="list-style-type: none"> • Find issues that require or required approval by John Smith: <code>approval = approver(jsmith)</code> • Find issues that require approval by John Smith: <code>approval = pendingBy(jsmith)</code> • Find issues that require approval by the current user: <code>approval = myPending()</code> • Find all issues that require approval: <code>approval = pending()</code>

[^ top of page](#)

Assignee

Search for issues that are assigned to a particular user. You can search by the user's full name, ID, or email address.

Syntax	<code>assignee</code>
Alias	<code>cf[CustomFieldID]</code>
Field Type	USER
Auto-complete	Yes

Supported operators	<p>= , != IS, IS NOT, IN, NOT IN, WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED</p> <p><i>Note that the comparison operators (e.g. ">") use the version order that has been set up by your project administrator, not a numeric or alphabetic order.</i></p>
Unsupported operators	<p>~ , !~ , > , >= , < , <=</p>
Supported functions	<p>When used with the IN and NOT IN operators, this field supports:</p> <ul style="list-style-type: none"> membersOf() <p>When used with the EQUALS and NOT EQUALS operators, this field supports:</p> <ul style="list-style-type: none"> currentUser()
Examples	<ul style="list-style-type: none"> Find issues that are assigned to John Smith: assignee = "John Smith" or assignee = jsmith Find issues that are currently assigned, or were previously assigned, to John Smith: assignee WAS "John Smith" or assignee WAS jsmith Find issues that are assigned by the user with email address "bob@mycompany.com": assignee = "bob@mycompany.com" <p><i>Note that full-stops and "@" symbols are reserved characters and need to be surrounded by quote-marks.</i></p>

[^ top of page](#)

Attachments

Search for issues that have or do not have attachments.

Syntax	attachments
Field Type	ATTACHMENT
Auto-complete	Yes
Supported operators	IS, IS NOT
Unsupported operators	=, !=, ~, !~, >, >=, <, <= IN, NOT IN, WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED
Supported functions	None

Examples	<ul style="list-style-type: none"> • Search for issues that have attachments: attachments IS NOT EMPTY • Search for issues that do not have attachments: attachments IS EMPTY
-----------------	---

[^ top of page](#)

Category

Search for issues that belong to projects in a particular category.

Syntax	category
Field Type	CATEGORY
Auto-complete	Yes
Supported operators	=, != IS, IS NOT, IN, NOT IN
Unsupported operators	~, !~, >, >=, <, <= WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED
Supported functions	None
Examples	<ul style="list-style-type: none"> • Find issues that belong to projects in the "Alphabet Projects" Category: category = "Alphabet Projects"

[^ top of page](#)

Comment

Search for issues that have a comment that contains particular text. [Jira text-search syntax](#) can be used.

Syntax	comment
Field Type	TEXT
Auto-complete	No
Supported operators	~, !~
Unsupported operators	=, !=, >, >=, <, <= IS, IS NOT, IN, NOT IN, WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED
Supported functions	None

Examples	<ul style="list-style-type: none"> Find issues where a comment contains text that matches "My PC is quite old" (i.e. a "fuzzy" match): comment ~ "My PC is quite old" Find issues where a comment contains the exact phrase "My PC is quite old": comment ~ "\"My PC is quite old\""
-----------------	--

[^ top of page](#)

Component

Search for issues that belong to a particular component(s) of a project. You can search by component name or component ID (i.e. the number that Jira automatically allocates to a component).

Note, it is safer to search by component ID than by component name. Different projects may have components with the same name, so searching by component name may return issues from multiple projects. It is also possible for your Jira administrator to change the name of a component, which could break any saved filters that rely on that name. Component IDs, however, are unique and cannot be changed.

Syntax	component
Field Type	COMPONENT
Auto-complete	Yes
Supported operators	= , != IS , IS NOT , IN , NOT IN
Unsupported operators	~ , !~ , > , >= , < , <= WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Supported functions	When used with the IN and NOT IN operators, component supports: <ul style="list-style-type: none"> componentsLeadByUser()
Examples	<ul style="list-style-type: none"> Find issues in the "Comp1" or "Comp2" component: component in (Comp1, Comp2) Find issues in the "Comp1" and "Comp2" components: component in (Comp1) and component in (Comp2) or component = Comp1 and component = Comp2 Find issues in the component with ID 20500: component = 20500

[^ top of page](#)

Created

Search for issues that were created on, before, or after a particular date (or date range). Note that if a time-component is not specified, midnight will be assumed. Please note that the search results will be relative to your configured time zone (which is by default the Jira server's time zone).

Use one of the following formats:

"YYYY/MM/dd HH:mm"
 "YYYY-MM-dd HH:mm"
 "YYYY/MM/dd"
 "YYYY-MM-dd"

Or use "w" (weeks), "d" (days), "h" (hours) or "m" (minutes) to specify a date relative to the current time. The default is "m" (minutes). Be sure to use quote-marks (""); if you omit the quote-marks, the number you supply will be interpreted as milliseconds after epoch (1970-1-1).

Syntax	created
Alias	createdDate
Field Type	DATE
Auto-complete	No
Supported operators	= , != , > , >= , < , <= IS , IS NOT , IN , NOT IN
Unsupported operators	~ , !~ WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Supported functions	When used with the EQUALS, NOT EQUALS, GREATER THAN, GREATER THAN EQUALS, LESS THAN or LESS THAN EQUALS operators, this field supports: <ul style="list-style-type: none"> • currentLogin() • lastLogin() • now() • startOfDay() • startOfWeek() • startOfMonth() • startOfYear() • endOfDay() • endOfWeek() • endOfMonth() • endOfYear()
Examples	<ul style="list-style-type: none"> • Find all issues created before 12th December 2010: created < "2010/12/12" • Find all issues created on or before 12th December 2010: created <= "2010/12/13" • Find all issues created on 12th December 2010 before 2:00pm: created > "2010/12/12"and created < "2010/12/12 14:00" • Find issues created less than one day ago: created > "-1d" • Find issues created in January 2011: created > "2011/01/01"and created < "2011/02/01" • Find issues created on 15 January 2011: created > "2011/01/15"and created < "2011/01/16"

[^ top of page](#)

Creator

Search for issues that were created by a particular user. You can search by the user's full name, ID, or email address.

Syntax	<code>creator</code>
Field Type	USER
Auto-complete	Yes
Supported operators	<code>=</code> , <code>!=</code> <code>IS</code> , <code>IS NOT</code> , <code>IN</code> , <code>NOT IN</code> , <code>WAS</code> , <code>WAS IN</code> , <code>WAS NOT</code> , <code>WAS NOT IN</code>
Unsupported operators	<code>~</code> , <code>!~</code> , <code>></code> , <code>>=</code> , <code><</code> , <code><=</code> <code>CHANGED</code>
Supported functions	When used with the IN and NOT IN operators, this field supports: <ul style="list-style-type: none"> membersOf() When used with the EQUALS and NOT EQUALS operators, this field supports: <ul style="list-style-type: none"> currentUser()
Examples	<ul style="list-style-type: none"> Search for issues that were created by Jill Jones: <code>creator = "Jill Jones"</code> or <code>creator = "jjones"</code> Search for issues that were created by the user with email address "bob@mycompany.com": <code>creator = "bob@mycompany.com"</code> <p><i>(Note that full-stops and "@" symbols are reserved characters, so the email address needs to be surrounded by quote-marks.)</i></p>

[^ top of page](#)

Custom field

Only applicable if your Jira administrator has created one or more custom fields.

Search for issues where a particular custom field has a particular value. You can search by custom fieldname or custom fieldID (i.e. the number that Jira automatically allocates to an custom field).

Note, it is safer to search by custom fieldID than by custom fieldname. It is possible for a custom field to have the same name as a built-in Jira system field; in which case, Jira will search for the system field (not your custom field). It is also possible for your Jira administrator to change the name of a custom field, which could break any saved filters that rely on that name. Custom fieldIDs, however, are unique and cannot be changed.

Syntax	<code>CustomFieldName</code>
Alias	<code>cf[CustomFieldID]</code>
Field Type	Depends on the custom field's configuration <i>Note, Jira text-search syntax can be used with custom fields of type 'Text'.</i>

Auto-complete	Yes, for custom fields of type picker, group picker, select, checkbox and radio button fields
Supported operators	<i>Different types of custom field support different operators.</i>
Supported operators: number and date fields	= , != , > , >= , < . <= IS , IS NOT , IN , NOT IN
Unsupported operators: number and date fields	~ , !~ WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Supported operators: picker, select, checkbox and radio button fields	= , != IS , IS NOT , IN , NOT IN
Unsupported operators: picker, select, checkbox and radio button fields	~ , !~ , > , >= , < . <= WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Supported operators: text fields	~ , !~ IS , IS NOT
Unsupported operators: text fields	= , != , > , >= , < . <= IN , NOT IN , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Unsupported operators	~ , !~ , > , >= , < , <= WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Supported functions	<i>Different types of custom fields support different functions.</i>
Supported functions: date/time fields	When used with the EQUALS, NOT EQUALS, GREATER THAN, GREATER THAN EQUALS, LESS THAN or LESS THAN EQUALS operators, this field supports: <ul style="list-style-type: none"> • currentLogin() • lastLogin() • now() • startOfDay() • startOfWeek() • startOfMonth() • startOfYear() • endOfDay() • endOfWeek() • endOfMonth() • endOfYear()


<p>Supported functions: version picker fields</p>	<p>Version picker fields: When used with the IN and NOT IN operators, this field supports:</p> <ul style="list-style-type: none"> releasedVersions() latestReleasedVersion() unreleasedVersions() earliestUnreleasedVersion()
<p>Examples</p>	<ul style="list-style-type: none"> Find issues where the value of the "Location" custom field is "New York": location = "New York" Find issues where the value of the custom field with ID 10003 is "New York": cf[10003] = "New York" Find issues where the value of the "Location" custom field is "London" or "Milan" or "Paris": cf[10003] in ("London" , "Milan" , "Paris") Find issues where the "Location" custom field has no value: location != empty

[^ top of page](#)

Customer Request Type

Only applicable if Jira Service Management is installed and licensed.

Search for Issues matching a specific Customer Request Type in a service desk project. You can search for a Customer Request Type either by name or description as configured in the Request Type configuration screen.

<p>Syntax</p>	<p>"Customer Request Type"</p>
<p>Field Type</p>	<p>Custom field</p>
<p>Auto-complete</p>	<p>Yes</p>
<p>Supported operators</p>	<p style="text-align: center;">= , !=</p> <p>IN , NOT IN</p>
<p>Unsupported operators</p>	<p>~ , !~ , > , >= , < , <=</p> <p>IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p> Note that the Lucene value for Customer Request Type, is portal-key /request-type-key . While the portal key cannot be changed after a service desk portal is created, the project key can be changed. The Request Type key cannot be changed once the Request Type is created.</p> </div>
<p>Supported functions</p>	<p>None</p>

Examples	<ul style="list-style-type: none"> Find issues where Customer Request Type is Request a new account in projects that the user has access to: "Customer Request Type" = "Request a new account" Find issues where the Customer Request Type is Request a new account in SimpleDesk project, where the right operand is a selected Lucene value from the auto-complete suggestion list. "Customer Request Type" = "sd/system-access" Find issues where Customer Request Type is either Request a new account or Get IT Help. "Customer Request Type" IN("Request a new account", "Get IT Help")
-----------------	---

[^ top of page](#)

Description

Search for issues where the description contains particular text. [Jira text-search syntax](#) can be used.

Syntax	description
Field Type	TEXT
Auto-complete	No
Supported operators	~ , !~ IS , IS NOT
Unsupported operators	= , != , > , >= , < , <= IN , NOT IN , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Supported functions	None
Examples	<ul style="list-style-type: none"> Find issues where the description contains text that matches "Please see screenshot" (i.e. a "fuzzy" match): description ~ "Please see screenshot" Find issues where the description contains the exact phrase "Please see screenshot": description ~ "\"Please see screenshot\""

[^ top of page](#)

Due

Search for issues that were due on, before, or after a particular date (or date range). Note that the due date relates to the *date* only (not to the time).

Use one of the following formats:

"yyyy/MM/dd"
"yyyy-MM-dd"

Or use "w" (weeks) or "d" (days) to specify a date relative to the current date. Be sure to use quote-marks (").

Syntax	due
---------------	-----

Alias	dueDate
Field Type	DATE
Auto-complete	No
Supported operators	= , != , > , >= , < , <= IS , IS NOT , IN , NOT IN
Unsupported operators	~ , !~ WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Supported functions	When used with the EQUALS, NOT EQUALS, GREATER THAN, GREATER THAN EQUALS, LESS THAN or LESS THAN EQUALS operators, this field supports: <ul style="list-style-type: none"> • currentLogin() • lastLogin() • now() • startOfDay() • startOfWeek() • startOfMonth() • startOfYear() • endOfDay() • endOfWeek() • endOfMonth() • endOfYear()
Examples	<ul style="list-style-type: none"> • Find all issues due before 31st December 2010: due < "2010/12/31" • Find all issues due on or before 31st December 2010: due <= "2011/01/01" • Find all issues due tomorrow: due = "1d" • Find all issues due in January 2011: due >= "2011/01/01" and due <= "2011/01/31" • Find all issues due on 15 January 2011: due = "2011/01/15"

[^ top of page](#)

Environment

Search for issues where the environment contains particular text. [Jira text-search syntax](#) can be used.

Syntax	environment
Field Type	TEXT
Auto-complete	No
Supported operators	~ , !~ IS , IS NOT
Unsupported operators	= , != , > , >= , < , <= IN , NOT IN , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED

Supported functions	None
Examples	<ul style="list-style-type: none"> Find issues where the environment contains text that matches "Third floor" (i.e. a "fuzzy" match): environment ~ "Third floor" Find issues where the environment contains the exact phrase "Third floor": environment ~ "\"Third floor\""

[^ top of page](#)

Epic link

Search for issues that belong to a particular epic. The search is based on either the epic's name, issue key, or issue ID (i.e. the number that Jira automatically allocates to an issue).

Syntax	"epic link"
Field Type	Epic Link Relationship
Auto-complete	No
Supported operators	= , != IS , IS NOT, IN , NOT IN
Unsupported operators	~ , !~ , > , >= , < , <= WAS, WAS IN, WAS NOT, WAS NOT IN , CHANGED
Supported functions	When used with the IN or NOT IN operators, <code>epic link</code> supports: <ul style="list-style-type: none"> issueHistory() linkedIssues() votedIssues() watchedIssues()
Examples	<ul style="list-style-type: none"> Find issues that belong to epic "Jupiter", where "Jupiter" has the issue key ANERDS-31: "epic link"= ANERDS- 31 or "epic link"= Jupiter

[^ top of page](#)

Filter

You can use a saved filter to narrow your search. You can search by filter name or filter ID (i.e. the number that Jira automatically allocates to a saved filter).

Note:

- It is safer to search by filter ID than by filter name. It is possible for a filter name to be changed, which could break a saved filter that invokes another filter by name. Filter IDs, however, are unique and cannot be changed.
- An unnamed link statement in your typed query will override an ORDER BY statement in the saved filter.

- You cannot run or save a filter that would cause an infinite loop (i.e. you cannot reference a saved filter if it eventually references your current filter).

Syntax	<code>filter</code>
Aliases	<code>request</code> , <code>savedFilter</code> , <code>searchRequest</code>
Field Type	Filter
Auto-complete	Yes
Supported operators	<code>=</code> , <code>!=</code> <code>IN</code> , <code>NOT IN</code>
Unsupported operators	<code>~</code> , <code>!~</code> , <code>></code> , <code>>=</code> , <code><</code> , <code><=</code> <code>IS</code> , <code>IS NOT</code> , <code>WAS</code> , <code>WAS IN</code> , <code>WAS NOT</code> , <code>WAS NOT IN</code> , <code>CHANGED</code>
Supported functions	None
Examples	<ul style="list-style-type: none"> Search the results of the filter "My Saved Filter" (which has an ID of 12000) for issues assigned to the user jsmith: <code>filter = "My Saved Filter"and assignee = jsmith</code> or <code>filter = 12000and assignee = jsmith</code>

[^ top of page](#)

Fix version

Search for issues that are assigned to a particular fix version. You can search by version name or version ID (i.e. the number that Jira automatically allocates to a version).

Note, it is safer to search by version ID than by version name. Different projects may have versions with the same name, so searching by version name may return issues from multiple projects. It is also possible for your Jira administrator to change the name of a version, which could break any saved filters that rely on that name. Version IDs, however, are unique and cannot be changed.

Syntax	<code>fixVersion</code>
Field Type	VERSION
Auto-complete	Yes
Supported operators	<code>=</code> , <code>!=</code> , <code>></code> , <code>>=</code> , <code><</code> , <code><=</code> , <code>~</code> , <code>!~</code> <code>IS</code> , <code>IS NOT</code> , <code>IN</code> , <code>NOT IN</code> , <code>WAS</code> , <code>WAS IN</code> , <code>WAS NOT</code> , <code>WAS NOT IN</code> , <code>CHANGED</code> <ul style="list-style-type: none"> The comparison operators (e.g. ">") use the version order that has been set up by your project administrator, not a numeric or alphabetic order. For this field, the contain operators (~ and !~) find exact matches, and can be used to search through versions with a wildcard.
Unsupported operators	

Supported functions	<p>When used with the IN and NOT IN operators, this field supports:</p> <ul style="list-style-type: none"> releasedVersions() latestReleasedVersion() unreleasedVersions() earliestUnreleasedVersion()
Examples	<ul style="list-style-type: none"> Find issues with a Fix Version of 3.14 or 4.2: <code>fixVersion in ("3.14" , "4.2")</code> <i>(Note that full-stops are reserved characters, so they need to be surrounded by quote-marks.)</i> Find issues with a Fix Version of "Little Ted": <code>fixVersion = "Little Ted"</code> Find issues with a Fix Version ID of 10001: <code>fixVersion = 10001</code>

[^ top of page](#)

Issue key

Search for issues with a particular issue key or issueID (i.e. the number that Jira automatically allocates to an issue).


Syntax	<code>issueKey</code>
Aliases	<code>id , issue , key</code>
Field Type	ISSUE
Auto-complete	No
Supported operators	<code>= , != , > , >= , < , <=</code> <code>IS , IS NOT , IN , NOT IN</code>
Unsupported operators	<code>~ , !~</code> <code>WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Supported functions	<p>When used with the IN or NOT IN operators, <code>issueKey</code> supports:</p> <ul style="list-style-type: none"> issueHistory() linkedIssues() updatedBy() votedIssues() watchedIssues()
Examples	<ul style="list-style-type: none"> Find the issue with key "ABC-123": <code>issueKey = ABC- 123</code>


[^ top of page](#)

Issue link type

Issue linking allows you to create associations between issues on either the same or different Jira servers. For example, an issue may *duplicate* another issue or *depend on* the resolution of another issue. You can find detailed information about issue links in [Configuring issue linking](#).

When searching for issues with a particular link type, you can only find linked issues that are on the same Jira instance you're searching on. Links to issues on a remote Jira instance or to Confluence pages won't be included.

 Use the following JQL query to add colors to your issue cards! For example, add a red stripe to issues that have some blockers, and keep all other issues green. This will help you bring the right information to your teams attention, at a glance. For more info, see [Customizing cards](#).

Syntax	issueLinkType
Auto-complete	Yes
Supported operators	= , != IN , NOT IN
Unsupported operators	~ , !~ , > , >= , < , <= WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED , IS , IS NOT
Supported functions	None
Examples	<ul style="list-style-type: none"> Find issues with a link type of "blocks": <code>issueLinkType = blocks</code> Find issues with an issue type of "duplicates" or "is duplicated by": <code>issueLinkType in (duplicates, "is duplicated by")</code> Find issues with link types other than clones: <code>issueLinkType != clones</code>  This query will also return issues with no links at all. Find issues that are blocker by other issues, or that don't have any blockers. <code>issueLinkType = "is blocked by"</code> <code>issueLinkType != "is blocked by"</code>

[^ top of page](#)

Labels

Search for issues tagged with a label or list of labels. You can also search for issues without any labels to easily identify which issues need to be tagged so they show up in the relevant sprints, queues or reports.

Syntax	labels
Field Type	LABEL
Auto-complete	Yes
Supported operators	= , !=, IS, IS NOT, IN, NOT IN <i>We recommend using IS or IS NOT to search for a single label, and IN or NOT IN to search for a list of labels.</i>
Unsupported operators	~ , !~ , , > , >= , < , <= WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED

Supported functions	None
Examples	<ul style="list-style-type: none"> Find issues with an existing label: labels = "x" Find issues without a specified label, including issues without a label: labels not in ("x") or labels is EMPTY

Last viewed

Search for issues that were last viewed on, before, or after a particular date (or date range). Note that if a time-component is not specified, midnight will be assumed. Please note that the search results will be relative to your configured time zone (which is by default the Jira server's time zone).

Use one of the following formats:

"YYYY/MM/dd HH:mm"
 "YYYY-MM-dd HH:mm"
 "YYYY/MM/dd"
 "YYYY-MM-dd"

Or use "w" (weeks), "d" (days), "h" (hours) or "m" (minutes) to specify a date relative to the current time. The default is "m" (minutes). Be sure to use quote-marks (""); if you omit the quote-marks, the number you supply will be interpreted as milliseconds after epoch (1970-1-1).

Syntax	lastViewed
Field Type	DATE
Auto-complete	No
Supported operators	= , != , > , >= , < , <= IS , IS NOT, IN , NOT IN
Unsupported operators	~ , !~ WAS, WAS IN, WAS NOT, WAS NOT IN , CHANGED
Supported functions	When used with the EQUALS,NOT EQUALS,GREATER THAN,GREATER THAN EQUALS,LESS THAN or LESS THAN EQUALS operators, this field supports: <ul style="list-style-type: none"> currentLogin() lastLogin() now() startOfDay() startOfWeek() startOfMonth() startOfYear() endOfDay() endOfWeek() endOfMonth() endOfYear()

Examples	<ul style="list-style-type: none"> Find all issues last viewed before 12th December 2010: lastViewed < "2010/12/12" Find all issues last viewed on or before 12th December 2010: lastViewed <= "2010/12/13" Find all issues last viewed on 12th December 2010 before 2:00pm: lastViewed > "2010/12/12"and created < "2010/12/12 14:00" Find issues last viewed less than one day ago: lastViewed > "-1d" Find issues last viewed in January 2011: lastViewed > "2011/01/01"and created < "2011/02/01" Find issues last viewed on 15 January 2011: lastViewed > "2011/01/15"and created < "2011/01/16"
-----------------	--

[^ top of page](#)

Level

Only available if issue level security has been enabled by your Jira administrator.

Search for issues with a particular security level. You can search by issue level securityname or issue level securityID (i.e. the number that Jira automatically allocates to an issue level security).

Note, it is safer to search by security levelID than by security level name.It is possible for your Jira administrator to change the name of a security level, which could break any saved filter that rely on that name. Security level IDs, however, are unique and cannot be changed.

Syntax	level
Field Type	SECURITY LEVEL
Auto-complete	Yes
Supported operators	= , != IS , IS NOT, IN , NOT IN
Unsupported operators	> , >= , < , <= , ~ , !~ WAS , WAS IN, WAS NOT, WAS NOT IN , CHANGED
Supported functions	None
Examples	<ul style="list-style-type: none"> Search for issues with a security level of "Really High" or "level1": level in ("Really High" , level1) Search for issues with a security level ID of 123: level = 123

[^ top of page](#)

Original estimate

Only available if time-tracking has been enabled by your Jira administrator.

Search for issues where the original estimate is set to a particular value (i.e. a number, not a date or date range).Use "w", "d", "h" and "m" to specify weeks, days, hours, or minutes.

Syntax	originalEstimate
---------------	------------------

Alias	timeOriginalEstimate
Field Type	DURATION
Auto-complete	No
Supported operators	= , != , > , >= , < , <= IS , IS NOT, IN , NOT IN
Unsupported operators	~ , !~ WAS , WAS IN, WAS NOT, WAS NOT IN , CHANGED
Supported functions	None
Examples	<ul style="list-style-type: none"> Find issues with an original estimate of 1 hour: originalEstimate = 1h Find issues with an original estimate of more than 2 days: originalEstimate > 2d

[^ top of page](#)

Parent

Only available if sub-tasks have been enabled by your Jira administrator.

Search for all sub-tasks of a particular issue. You can search by issue key or by issue ID (i.e. the number that Jira automatically allocates to an Issue).

Syntax	parent
Field Type	ISSUE
Auto-complete	No
Supported operators	= , != IN , NOT IN
Unsupported operators	> , >= , < , <= , ~ , !~ IS , IS NOT, WAS , WAS IN, WAS NOT, WAS NOT IN , CHANGED
Supported functions	None
Examples	<ul style="list-style-type: none"> Find issues that are sub-tasks of issue TEST-1234: parent = TEST- 1234

[^ top of page](#)

Priority

Search for issues with a particular priority. You can search by priorityname or priorityID (i.e. the number that Jira automatically allocates to a priority).

Note, it is safer to search by priorityID than by priorityname. It is possible for your Jira administrator to change the name of a priority, which could break any saved filter that rely on that name. Priority IDs, however, are unique and cannot be changed.

Syntax	<code>priority</code>
Field Type	PRIORITY
Auto-complete	Yes
Supported operators	<code>= , != , > , >= , < , <=</code> <code>IS , IS NOT , IN , NOT IN , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Unsupported operators	<code>~ , !~</code>
Supported functions	None
Examples	<ul style="list-style-type: none"> Find issues with a priority of "High": <code>priority = High</code> Find issues with a priority ID of 10000: <code>priority = 10000</code>

[^ top of page](#)

Project

Search for issues that belong to a particular project. You can search by `projectname`, by `projectkey` or by `projectID` (i.e. the number that Jira automatically allocates to a project). In the rare case where there is a project whose project key is the same as another project's name, then the project key takes preference and hides results from the second project.

Syntax	<code>project</code>
Field Type	PROJECT
Auto-complete	Yes
Supported operators	<code>= , !=</code> <code>IS , IS NOT , IN , NOT IN</code>
Unsupported operators	<code>> , >= , < , <= , ~ , !~</code> <code>WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Supported functions	When used with the <code>IN</code> and <code>NOT IN</code> operators, <code>project</code> supports: <ul style="list-style-type: none"> <code>projectsLeadByUser()</code> <code>projectsWhereUserHasPermission()</code> <code>projectsWhereUserHasRole()</code>
Examples	<ul style="list-style-type: none"> Find issues that belong to the Project that has the name "ABC Project": <code>project = "ABC Project"</code> Find issues that belong to the project that has the key "ABC": <code>project = "ABC"</code> Find issues that belong to the project that has the ID "1234": <code>project = 1234</code>

[^ top of page](#)

Remaining estimate

Only available if time-tracking has been enabled by your Jira administrator.

Search for issues where the remaining estimate is set to a particular value (i.e. a number, not a date or date range). Use "w", "d", "h" and "m" to specify weeks, days, hours, or minutes.

Syntax	<code>remainingEstimate</code>
Alias	<code>timeEstimate</code>
Field Type	DURATION
Auto-complete	No
Supported operators	<code>= , != , > , >= , < , <=</code> <code>IS , IS NOT , IN , NOT IN</code>
Unsupported operators	<code>~ , !~</code> <code>WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Supported functions	None
Examples	<ul style="list-style-type: none"> Find issues with a remaining estimate of more than 4 hours: <code>remainingEstimate > 4h</code>

[^ top of page](#)

Reporter

Search for issues that were reported by a particular user. This may be the same as the creator, but can be distinct. You can search by the user's full name, ID, or email address.

Syntax	<code>reporter</code>
Field Type	USER
Auto-complete	Yes
Supported operators	<code>= , !=</code> <code>IS , IS NOT , IN , NOT IN , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Unsupported operators	<code>~ , !~ , > , >= , < , <=</code>
Supported functions	<p>When used with the IN and NOT IN operators, this field supports:</p> <ul style="list-style-type: none"> <code>membersOf()</code> <p>When used with the EQUALS and NOT EQUALS operators, this field supports:</p> <ul style="list-style-type: none"> <code>currentUser()</code>

Examples	<ul style="list-style-type: none"> Search for issues that were reported by Jill Jones: <code>reporter = "Jill Jones"</code> or <code>reporter = jjones</code> Search for issues that were reported by the user with email address " bob@mycompany.com ": <code>reporter = "bob@mycompany.com"</code> (Note that full-stops and "@" symbols are reserved characters , so the email address needs to be surrounded by quote-marks.)
-----------------	--

[^ top of page](#)

Request channel type

Only applicable if Jira Service Management is installed and licensed.

Search for issues that were requested through a specific channel (e.g. issues submitted via email or through a Service Desk portal).

Syntax	<code>request-channel-type</code>
Field Type	TEXT
Auto-complete	Yes
Supported operators	<code>= , !=</code> <code>IS, IS NOT, IN, NOT IN</code>
Unsupported operators	<code>~ , !~ , > , >= , < , <=</code> <code>WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED</code>
Supported functions	When used with the IN and NOT IN operators, this field supports: <ul style="list-style-type: none"> <code>email</code>: requests submitted via email <code>Jira</code>: requests created using Jira <code>portal</code>: requests created using a Service Desk portal <code>api</code>: requests created using a REST API
Examples	<ul style="list-style-type: none"> Find issues where the request channel was email: <code>request-channel-type = email</code> Find issues where the request channel was something other than a service desk portal: <code>request-channel-type != portal</code>

[^ top of page](#)

Request last activity time

Only applicable if Jira Service Management is installed and licensed.

Search for issues that were last acted on or created:

- on a particular date.

- before/after a particular date (or date range).

Note that if a time-component is not specified, midnight will be assumed. Please note that the search results will be relative to your configured time zone (which is by default the Jira server's time zone).

Use one of the following formats:

```
"yyyy/MM/dd HH:mm"
"yyyy-MM-dd HH:mm"
"yyyy/MM/dd"
"yyyy-MM-dd"
```

Or use "w" (weeks), "d" (days), "h" (hours) or "m" (minutes) to specify a date relative to the current time. The default is "m" (minutes). Be sure to use quote-marks (" "); if you omit the quote-marks, the number you supply will be interpreted as milliseconds after epoch (1970-1-1).

Syntax	request-last-activity-time
Field Type	DATE
Auto-complete	Yes
Supported operators	= , != , > , >= , < , <= IS, IS NOT, IN, NOT IN
Unsupported operators	~ , !~ WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED
Supported functions	When used with the EQUALS, NOT EQUALS, GREATER THAN, GREATER THAN EQUALS, LESS THAN or LESS THAN EQUALS operators, this field supports: <ul style="list-style-type: none"> • currentLogin() • lastLogin() • now() • startOfDay() • startOfWeek() • startOfMonth() • startOfYear() • endOfDay() • endOfWeek() • endOfMonth() • endOfYear()
Examples	<ul style="list-style-type: none"> • Find all issues last acted on before 23rd May 2016: request-last-activity-time < "2016/05/23" • Find all issues last acted on or before 23rd May 2016: request-last-activity-time <= "2016/05/23" • Find all issues created on 23rd May 2016 and last acted on before 2:00pm that day: created > "2016/05/23" AND request-last-activity-time < "2016/05/23 14:00" • Find issues last acted on less than one day ago: request-last-activity-time > "-1d" • Find issues last acted on in January 2016: request-last-activity-time > "2016/01/01" and request-last-activity-time < "2016/02/01"

[^ top of page](#)

Resolution

Search for issues that have a particular resolution. You can search by resolutionname or resolutionID (i.e. the number that Jira automatically allocates to a resolution).

Note, it is safer to search by resolutionID than by resolutionname. It is possible for your Jira administrator to change the name of a resolution, which could break any saved filter that rely on that name. Resolution IDs, however, are unique and cannot be changed.

Syntax	resolution
Field Type	RESOLUTION
Auto-complete	Yes
Supported operators	= , != , > , >= , < , <= IS , IS NOT , IN , NOT IN , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Unsupported operators	~ , !~
Supported functions	None
Examples	<ul style="list-style-type: none"> Find issues with a resolution of "Cannot Reproduce" or "Won't Fix": resolution in ("Cannot Reproduce", "Won't Fix") Find issues with a resolution ID of 5: resolution = 5 Find issues that do not have a resolution: resolution = unresolved

[^ top of page](#)

Resolved

Search for issues that were resolved on, before, or after a particular date (or date range). Note that if a time-component is not specified, midnight will be assumed. Please note that the search results will be relative to your configured time zone (which is by default the Jira server's time zone).

Use one of the following formats:

```
"YYYY/MM/dd HH:mm"  
"YYYY-MM-dd HH:mm"  
"YYYY/MM/dd"  
"YYYY-MM-dd"
```

Or use "w" (weeks), "d" (days), "h" (hours) or "m" (minutes) to specify a date relative to the current time. The default is "m" (minutes). Be sure to use quote-marks (" "); if you omit the quote-marks, the number you supply will be interpreted as milliseconds after epoch (1970-1-1).

Syntax	resolved
Alias	resolutionDate

Field Type	DATE
Auto-complete	No
Supported operators	= , != , > , >= , < , <= IS , IS NOT, IN , NOT IN
Unsupported operators	~ , !~ WAS, WAS IN, WAS NOT, WAS NOT IN , CHANGED
Supported functions	When used with the EQUALS, NOT EQUALS, GREATER THAN, GREATER THAN EQUALS, LESS THAN or LESS THAN EQUALS operators, this field supports: <ul style="list-style-type: none"> • currentLogin() • lastLogin() • now() • startOfDay() • startOfWeek() • startOfMonth() • startOfYear() • endOfDay() • endOfWeek() • endOfMonth() • endOfYear()
Examples	<ul style="list-style-type: none"> • Find all issues that were resolved before 31st December 2010: <code>resolved <= "2010/12/31"</code> • Find all issues that were resolved before 2.00pm on 31st December 2010: <code>resolved < "2010/12/31 14:00"</code> • Find all issues that were resolved on or before 31st December 2010: <code>resolved <= "2011/01/01"</code> • Find issues that were resolved in January 2011: <code>resolved > "2011/01/01" and resolved < "2011/02/01"</code> • Find issues that were resolved on 15 January 2011: <code>resolved > "2011/01/15" and resolved < "2011/01/16"</code> • Find issues that were resolved in the last hour: <code>resolved > -1h</code>

[^ top of page](#)

SLA

Used in Jira Service Management only

Search for requests whose SLAs are in a certain

Syntax	Time to resolution Time to first response <your custom SLA name>
Field Type	SLA
Auto-complete	No
Supported operators	= , !=, > , >= , < , <=

Unsupported operators	<p>~ , !~ IS , IS NOT , IN , NOT IN , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</p>
Supported functions	<ul style="list-style-type: none"> • breached() • completed() • elapsed() • everBreached() • paused() • remaining() • running() • withinCalendarHours()
Examples	<ul style="list-style-type: none"> • Find issues where Time to First Response was breached: "Time to First Response" = everBreached() • Find issues where the SLA for Time to Resolution is paused due to a condition: "Time to Resolution" = paused() • Find issues where the SLA for Time to Resolution is paused due to the SLA calendar: "Time to Resolution" = withinCalendarHours() • Find issues that have been waiting for a response for more than 1 hour: "Time to First Response" > elapsed("1h") • Find issues that that will breach Time to First Response in the next two hours: "Time to First Response" < remaining("2h")

[^ top of page](#)

Sprint

Search for issues that are assigned to a particular sprint. This works for active sprints and future sprints. The search is based on either the sprint name or the sprint ID (i.e. the number that Jira automatically allocates to a sprint).

If you have multiple sprints with similar (or identical) names, you can simply search by using the sprint name or even just part of it. The possible matches will be shown in the autocomplete drop-down, with the sprint dates shown to help you distinguish between them. (The sprint ID will also be shown, in brackets).

Syntax	sprint
Field Type	NUMBER
Auto-complete	Yes
Supported operators	<p>= , != IS , IS NOT , IN , NOT IN</p>
Unsupported operators	<p>~ , !~ , > , >= , < , <= WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</p>
Supported functions	<ul style="list-style-type: none"> • openSprints() • closedSprints()

Examples	<ul style="list-style-type: none"> Find issues that belong to sprint 999: <code>sprint = 999</code> Find issues that belong to sprint "February 1": <code>sprint = "February 1"</code> Find issues that belong to either "February 1", "February 2" or "February 3": <code>sprint in ("February 1" , "February 2" , "February 3")</code> Find issues that are assigned to a sprint: <code>sprint is not empty</code>
-----------------	--

[^ top of page](#)

Status

Search for issues that have a particular status. You can search by statusname or statusID (i.e. the number that Jira automatically allocates to a status).

Note:

- It is safer to search by statusID than statusname. It is possible for your Jira administrator to change the name of a status, which could break any saved filter that rely on that name. Status IDs, however, are unique and cannot be changed.
- The WAS, WAS NOT, WAS IN and WAS NOT IN operators can only be used with the name, not the ID.

Syntax	<code>status</code>
Field Type	STATUS
Auto-complete	Yes
Supported operators	<code>= , !=</code> <code>IS , IS NOT , IN , NOT IN , WAS , WAS IN , WAS NOT , WAS NOT IN</code> <code> , CHANGED</code>
Unsupported operators	<code>~ , !~ , > , >= , < , <=</code>
Supported functions	None
Examples	<ul style="list-style-type: none"> Find issues with a status of "Open": <code>status = Open</code> Find issues with a status ID of 1: <code>status = 1</code> Find issues that currently have, or previously had, a status of "Open": <code>status WAS Open</code>

[^ top of page](#)

Summary

Search for issues where the summary contains particular text. [Jira text-search syntax](#) can be used.

Syntax	<code>summary</code>
---------------	----------------------

Field Type	TEXT
Auto-complete	No
Supported operators	~ , !~ IS , IS NOT
Unsupported operators	= , != , > , >= , < , <= IN , NOT IN , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Supported functions	None
Examples	<ul style="list-style-type: none"> Find issues where the summary contains text that matches "Error saving file" (i.e. a "fuzzy" match): summary ~ "Error saving file" Find issues where the summary contains the exact phrase "Error saving file": summary ~ "\"Error saving file\""

[^ top of page](#)

Text

This is a "master-field" that allows you to search all text fields, i.e.:

- Summary
- Description
- Environment
- Comments
- custom fields that use the "free text searcher"; this includes custom fields of the following built-in custom field types:
 - Free text field (unlimited text)
 - Textfield(< 255 characters)
 - Read-onlytext field

Notes:

- The `text` master-field can only be used with the `CONTAINS` operator ("`~`" and "`!~`").
- [Jira text-search syntax](#) can be used with these fields.

Syntax	text
Field Type	TEXT
Auto-complete	No
Supported operators	~
Unsupported operators	= , != , !~ , > , >= , < , <= IS , IS NOT , IN , NOT IN , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Supported functions	None

Examples	<ul style="list-style-type: none"> Find issues where a text field matches the word "Fred": <code>text ~ "Fred"</code> or <code>text ~ Fred</code> Find all issues where a text field contains the exact phrase "full screen": <code>text ~ "\"full screen\""</code>
-----------------	---

[^ top of page](#)

Time spent

Only available if time-tracking has been enabled by your Jira administrator.

Search for issues where the time spent is set to a particular value (i.e. a number, not a date or date range). Use "w", "d", "h" and "m" to specify weeks, days, hours, or minutes.

Syntax	<code>timeSpent</code>
Field Type	DURATION
Auto-complete	No
Supported operators	<code>= , != , > , >= , < , <=</code> <code>IS , IS NOT , IN , NOT IN</code>
Unsupported operators	<code>~ , !~</code> <code>WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Supported functions	None
Examples	<ul style="list-style-type: none"> Find issues where the time spent is more than 5 days: <code>timeSpent > 5d</code>

[^ top of page](#)

Type

Search for issues that have a particular issue type. You can search by issue typename or issue typeID (i.e. the number that Jira automatically allocates to an issue type).

Note, it is safer to search by typeID than typename. It is possible for your Jira administrator to change the name of a type, which could break any saved filter that rely on that name. Type IDs, however, are unique and cannot be changed.

Syntax	<code>type</code>
Alias	<code>issueType</code>
Field Type	ISSUE_TYPE
Auto-complete	Yes
Supported operators	<code>= , !=</code> <code>IS , IS NOT , IN , NOT IN</code>

Unsupported operators	~ , !~ , > , >= , < , <= WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Supported functions	None
Examples	<ul style="list-style-type: none"> Find issues with an issue type of "Bug": type = Bug Find issues with an issue type of "Bug" or "Improvement": issueType in (Bug,Improvement) Find issues with an issue type ID of 2: issueType = 2

[^ top of page](#)

Updated

Search for issues that were last updated on, before, or after a particular date (or date range). Note that if a time-component is not specified, midnight will be assumed. Please note that the search results will be relative to your configured time zone (which is by default the Jira server's time zone).

Use one of the following formats:

"YYYY/MM/dd HH:mm"
 "YYYY-MM-dd HH:mm"
 "YYYY/MM/dd"
 "YYYY-MM-dd"

Or use "w" (weeks), "d" (days), "h" (hours) or "m" (minutes) to specify a date relative to the current time. The default is "m" (minutes). Be sure to use quote-marks (""); if you omit the quote-marks, the number you supply will be interpreted as milliseconds after epoch (1970-1-1).

Syntax	updated
Alias	updatedAt
Field Type	DATE
Auto-complete	No
Supported operators	= , != , > , >= , < , <= IS , IS NOT , IN , NOT IN
Unsupported operators	~ , !~ WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED

<p>Supported functions</p>	<p>When used with the EQUALS, NOT EQUALS, GREATER THAN, GREATER THAN EQUALS, LESS THAN or LESS THAN EQUALS operators, this field supports:</p> <ul style="list-style-type: none"> • <code>currentLogin()</code> • <code>lastLogin()</code> • <code>now()</code> • <code>startOfDay()</code> • <code>startOfWeek()</code> • <code>startOfMonth()</code> • <code>startOfYear()</code> • <code>endOfDay()</code> • <code>endOfWeek()</code> • <code>endOfMonth()</code> • <code>endOfYear()</code>
<p>Examples</p>	<ul style="list-style-type: none"> • Find issues that were last updated before 12th December 2010: <code>updated < "2010/12/12"</code> • Find issues that were last updated on or before 12th December 2010: <code>updated < "2010/12/13"</code> • Find all issues that were last updated before 2.00pm on 31st December 2010: <code>updated < "2010/12/31 14:00"</code> • Find issues that were last updated more than two weeks ago: <code>updated < "-2w"</code> • Find issues that were last updated on 15 January 2011: <code>updated > "2011/01/15" and updated < "2011/01/16"</code> • Find issues that were last updated in January 2011: <code>updated > "2011/01/01" and updated < "2011/02/01"</code>

[^ top of page](#)

Voter

Search for issues for which a particular user has voted. You can search by the user's full name, ID, or email address. Note that you can only find issues for which you have the "View Voters and Watchers" permission, unless you are searching for your own votes. See also [votedIssues](#).

<p>Syntax</p>	<p><code>voter</code></p>
<p>Field Type</p>	<p>USER</p>
<p>Auto-complete</p>	<p>Yes</p>
<p>Supported operators</p>	<p><code>= , !=</code> <code>IS , IS NOT , IN , NOT IN</code></p>
<p>Unsupported operators</p>	<p><code>~ , !~, > , >= , < , <=</code> <code>WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code></p>
<p>Supported functions</p>	<p>When used with the IN and NOT IN operators, this field supports:</p> <ul style="list-style-type: none"> • <code>membersOf()</code> <p>When used with the EQUALS and NOT EQUALS operators, this field supports:</p> <ul style="list-style-type: none"> • <code>currentUser()</code>

Examples	<ul style="list-style-type: none"> • Search for issues that you have voted for: <code>voter = currentUser()</code> • Search for issues that the user "jsmith" has voted for: <code>voter = "jsmith"</code> • Search for issues for which a member of the group "Jira-administrators" has voted: <code>voter in membersOf("Jira-administrators")</code>
-----------------	---

[^ top of page](#)

Votes

Search for issues with a specified number of votes.

Syntax	<code>votes</code>
Field Type	NUMBER
Auto-complete	No
Supported operators	<code>= , != , > , >= , < , <=</code> <code>IN , NOT IN</code>
Unsupported operators	<code>~ , !~</code> <code>IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Supported functions	None
Examples	<ul style="list-style-type: none"> • Find all issues that have 12 or more votes: <code>votes >= 12</code>

[^ top of page](#)

Watcher

Search for issues that a particular user is watching. You can search by the user's full name, ID, or email address. Note that you can only find issues for which you have the "View Voters and Watchers" permission, unless you are searching for issues where you are the watcher. See also [watchedissues](#).

Syntax	<code>watcher</code>
Field Type	USER
Auto-complete	Yes
Supported operators	<code>= , !=</code> <code>IS , IS NOT , IN , NOT IN</code>
Unsupported operators	<code>~ , !~ , > , >= , < , <=</code> <code>WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>

<p>Supported functions</p>	<p>When used with the IN and NOT IN operators, this field supports:</p> <ul style="list-style-type: none"> membersOf() <p>When used with the EQUALS and NOT EQUALS operators, this field supports:</p> <ul style="list-style-type: none"> currentUser()
<p>Examples</p>	<ul style="list-style-type: none"> Search for issues that you are watching: watcher = currentUser() Search for issues that the user "jsmith" is watching: watcher = "jsmith" Search for issues that are being watched by a member of the group "Jira-administrators": watcher in membersOf("Jira-administrators")

[^ top of page](#)

Watchers

Search for issues with a specified number of watchers.

<p>Syntax</p>	<p>watchers</p>
<p>Field Type</p>	<p>NUMBER</p>
<p>Auto-complete</p>	<p>No</p>
<p>Supported operators</p>	<p>= , != , > , >= , < , <= IN , NOT IN</p>
<p>Unsupported operators</p>	<p>~ , !~ IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</p>
<p>Supported functions</p>	<p>When used with the IN and NOT IN operators, this field supports:</p> <ul style="list-style-type: none"> membersOf() <p>When used with the EQUALS and NOT EQUALS operators, this field supports:</p> <ul style="list-style-type: none"> currentUser()
<p>Examples</p>	<ul style="list-style-type: none"> Find all issues that are being watched by more than 3 people: watchers > 3

[^ top of page](#)

Work log author

Only available if time-tracking has been enabled by your Jira administrator.

Search for issues a particular user has logged work against. You can search by the user's full name, ID, or email address. Note that you can only find issues for which you have "Time Tracking" permissions, unless you are searching for issues that you've logged work against.

Syntax	worklogAuthor
Field Type	USER
Auto-complete	Yes
Supported operators	= , != IS , IS NOT , IN , NOT IN
Unsupported operators	~ , !~ , > , >= , < , <= WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Supported functions	When used with the IN and NOT IN operators, this field supports: <ul style="list-style-type: none"> membersOf() When used with the EQUALS and NOT EQUALS operators, this field supports: <ul style="list-style-type: none"> currentUser()
Examples	<ul style="list-style-type: none"> Search for issues that you've logged work against: worklogAuthor = currentUser() Search for issues that the user "jsmith" has logged work against: worklogAuthor = "jsmith" Search for issues that a member of the group "Jira-software-users": worklogAuthor in membersOf("Jira-software-users")

[^ top of page](#)

Work log comment

Only available if time-tracking has been enabled by your Jira administrator.

Search for issues that have a comment in a work log entry which contains particular text. [Jira text-search syntax](#) can be used.

Syntax	worklogComment
Field Type	TEXT
Auto-complete	No
Supported operators	~ , !~
Unsupported operators	= , != , > , >= , < , <= IS , IS NOT , IN , NOT IN , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Supported functions	None

Examples	<ul style="list-style-type: none"> Find issues where a comment in a work log entry contains text that matches "test sessions" (i.e. a "fuzzy" match): comment ~ "test sessions" Find issues where a comment contains the exact phrase "test sessions": summary ~ "\"test sessions\""
-----------------	--

[^ top of page](#)

Work log date

Only available if time-tracking has been enabled by your Jira administrator.

Search for issues that have comments in work log entries that were created on, before, or after a particular date (or date range). Note that if a time-component is not specified, midnight 00:00 will be assumed. Please note that the search results will be relative to your configured time zone (which is by default the Jira server's time zone).

Use one of the following formats:

"YYYY/MM/dd HH:mm"
 "YYYY-MM-dd HH:mm"
 "YYYY/MM/dd"
 "YYYY-MM-dd"

Or use "w" (weeks), "d" (days), "h" (hours) or "m" (minutes) to specify a date relative to the current time. The default is "m" (minutes). Be sure to use quote-marks (""); if you omit the quote-marks, the number you supply will be interpreted as milliseconds after epoch (1970-1-1).

Syntax	worklogDate
Field Type	DATE
Auto-complete	No
Supported operators	= , != , > , >= , < , <= IS , IS NOT , IN , NOT IN
Unsupported operators	~ , !~ WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Supported functions	When used with the EQUALS, NOT EQUALS, GREATER THAN, GREATER THAN EQUALS, LESS THAN or LESS THAN EQUALS operators, this field supports: <ul style="list-style-type: none"> currentLogin() lastLogin() now() startOfDay() startOfWeek() startOfMonth() startOfYear() endOfDay() endOfWeek() endOfMonth() endOfYear()

Examples	<ul style="list-style-type: none"> Find issues that have comments in work log entries created before midnight 00:00 12th December 2010: <code>worklogDate < "2010/12/12"</code> Find issues that have comments in work log entries created on or before 12th December 2010 (but not 13th December 2010): <code>worklogDate <= "2010/12/13"</code> Find issues that have comments in work log entries created on 12th December 2010 before 2:00pm: <code>worklogDate > "2010/12/12" and worklogDate < "2010/12/12 14:00"</code> Find issues that have comments in work log entries created less than one day ago: <code>worklogDate > "-1d"</code> Find issues that have comments in work log entries created in January 2011: <code>worklogDate > "2011/01/01" and worklogDate < "2011/02/01"</code> Find issues that have comments in work log entries created on 15 January 2011: <code>worklogDate > "2011/01/15" and worklogDate < "2011/01/16"</code>
-----------------	---

[^ top of page](#)

Work ratio

Only available if time-tracking has been enabled by your Jira administrator.

Search for issues where the work ratio has a particular value. Work ratio is calculated as follows: **workRatio = (timeSpent / originalEstimate) x 100**

Syntax	<code>workRatio</code>
Field Type	NUMBER
Auto-complete	No
Supported operators	<code>= , != , > , >= , < , <=</code> <code>IS , IS NOT , IN , NOT IN</code>
Unsupported operators	<code>~ , !~</code> <code>WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Supported functions	None
Examples	<ul style="list-style-type: none"> Find issues on which more than 75% of the original estimate has been spent: <code>workRatio > 75</code>

[^ top of page](#)

Advanced searching - keywords reference

This page describes information about keywords that are used for [advanced searching](#).

On this page:

- [AND](#)
- [OR](#)
- [NOT](#)
- [EMPTY](#)
- [NULL](#)
- [ORDER BY](#)

A keyword in JQL is a word or phrase that does any of the following:

- joins two or more clauses together to form a complex JQL query
- alters the logic of one or more clauses
- alters the logic of [operators](#)
- has an explicit definition in a JQL query
- performs a specific function that alters the results of a JQL query

AND

Used to combine multiple clauses, allowing you to refine your search.

You can also use parentheses to control the order in which clauses are executed. See [Precedence in JQL queries](#) for details.

- Find all open issues in the "New office" project:

```
project = "New office" and status = "open"
```

- Find all open, urgent issues that are assigned to jsmith:

```
status = open and priority = urgent and assignee = jsmith
```

- Find all issues in a particular project that are not assigned to jsmith:

```
project = JRA and assignee != jsmith
```

- Find all issues for a specific release which consists of different version numbers across several projects:

```
project in (JRA,CONF) and fixVersion = "3.14"
```

- Find all issues where neither the Reporter nor the Assignee is Jack, Jill or John:

```
reporter not in (Jack,Jill,John) and assignee not in (Jack,Jill,John)
```

OR

Used to combine multiple clauses, allowing you to expand your search.

You can also use parentheses to control the order in which clauses are executed. See [Precedence in JQL queries](#) for details.

- Find all issues that were created by either jsmith or jbrown:

```
reporter = jsmith or reporter = jbrown
```

- Find all issues that are overdue or where no due date is set:

```
duedate < now() or duedate is empty
```

- ✔ Check out the usage of [ofIN](#) operator, which can be a more convenient way to search for multiple values of a field.

NOT

Used to negate individual clauses or a complex JQL query (a query made up of more than one clause) using parentheses, allowing you to refine your search.

- Find all issues that are assigned to any user except jsmith:

```
not assignee = jsmith
```

- Find all issues that were not created by either jsmith or jbrown:

```
not (reporter = jsmith or reporter = jbrown)
```

- ✔ Check out the usage of [NOT EQUALS\("!="\)](#), [DOES NOT CONTAIN\("!~"\)](#), [NOT IN](#) and [IS NOT](#) operators that are often used to negate clauses in a JQL query.

EMPTY

Used to search for issues where a given field does not have a value. See also [NULL](#).

Note that EMPTY can only be used with fields that support the [IS](#) and [IS NOT](#) operators. To see a field's supported operators, check the individual [field](#) reference.

- Find all issues without a DueDate:

```
duedate = empty
```

or

```
duedate is empty
```

NULL

Used to search for issues where a given field does not have a value. See also [EMPTY](#).

Note that NULL can only be used with fields that support the [IS](#) and [IS NOT](#) operators. To see a field's supported operators, check the individual [field](#) reference.

- Find all issues without a DueDate:

```
duedate = null
```

or

```
duedate is null
```

ORDER BY

Used to specify the fields by whose values the search results will be sorted.

By default, the field's own sorting order will be used. You can override this by specifying ascending order ("asc") or descending order ("desc").

- Find all issues without aDueDate, sorted byCreationDate:


```
duedate = empty order by created
```

- Find all issues without aDueDate, sorted byCreationDate, then byPriority(highest to lowest):

```
duedate = empty order by created, priority desc
```

- Find all issues without aDueDate, sorted byCreationDate, then byPriority(lowest to highest):

```
duedate = empty order by created, priority asc
```

 Ordering by **Components** or **Versions** will list the returned issues first by **Project**, and only then by the field's natural order (see [JRA-31113](#)).

Advanced searching - operators reference


This page describes information about operators that are used for advanced searching.

An operator in JQL is one or more symbols or words that compare the value of a [field](#) on its left with one or more values or [functions](#) on its right. So, only valid results are retrieved by the clause. Some operators may use the **NOT** keyword.

EQUALS: =

The = operator is used to search for issues where the value of a specified field exactly matches a specified value.

To find issues where the value of a specified field exactly matches multiple values, use multiple EQUALS (=) statements with the [AND](#) keyword.

 The operator can't be used with text fields. See the [CONTAINS](#) operator.

Examples

- Find all issues that were created by jsmith:

```
reporter = jsmith
```


- Find all issues that were created by John Smith:

```
reporter = "John Smith"
```

[^top of page](#)

NOT EQUALS: !=

The != operator is used to search for issues where the value of a specified field doesn't match a specified value.



- The operator can't be used with text fields. See the [DOES NOT MATCH \(!~\)](#) operator instead.
- field != value is the same as NOT field = value.
- field != EMPTY is the same as field IS_NOT EMPTY.
- The operator won't match a field that has no value (an empty field). For example, component != fred will only match issues that have a component and this component isn't "fred". To find issues that have a component other than "fred" or have no component, you should type component != fred or component is empty.

Examples

- Find all issues that are assigned to any user except jsmith:

```
not assignee = jsmith
```

or

```
assignee != jsmith
```

On this page:

- [EQUALS: =](#)
- [NOT EQUALS: !=](#)
- [GREATER THAN: >](#)
- [GREATER THAN EQUALS: >=](#)
- [LESS THAN: <](#)
- [LESS THAN EQUALS: <=](#)
- [IN](#)
- [NOT IN](#)
- [CONTAINS: ~](#)
- [DOES NOT CONTAIN: !~](#)
- [IS](#)
- [IS NOT](#)
- [WAS](#)
- [WAS IN](#)
- [WAS NOT IN](#)
- [WAS NOT](#)
- [CHANGED](#)

- Find all issues that are not assigned to jsmith:

```
assignee != jsmith or assignee is empty
```

- Find all issues that were reported by you but aren't assigned to you:

```
reporter = currentUser() and assignee != currentUser()
```

- Find all issues where the Reporter or Assignee is anyone except John Smith:

```
assignee != "John Smith" or reporter != "John Smith"
```

- Find all issues that aren't unassigned:

```
assignee is not empty
```

or

```
assignee != null
```

[^top of page](#)

GREATER THAN: >

The > operator is used to search for issues where the value of a specified field is greater than a specified value.



The operator can only be used with fields that support ordering and can't be used with text fields. For example, date fields and version fields.

To see a field's supported operators, check the individual [field reference](#).

Examples

- Find all issues with more than four votes:

```
votes > 4
```

- Find all overdue issues:

```
duedate < now() and resolution is empty
```

- Find all issues where priority is higher than "Normal":

```
priority > normal
```

[^top of page](#)

GREATER THAN EQUALS: >=

The >= operator is used to search for issues where the value of a specified field is greater than or equal to a specified value.

i The operator can only be used with fields that support ordering and can't be used with text fields. For example, date fields and version fields.

To see a field's supported operators, check the individual [field reference](#).

Examples

- Find all issues with four or more votes:

```
votes >= 4
```

- Find all issues due on or after 31/12/2008:

```
duedate >= "2008/12/31"
```

- Find all issues created in the last five days:

```
created >= "-5d"
```

[^top of page](#)

LESS THAN: <

The < operator is used to search for issues where the value of a specified field is less than a specified value.

i The operator can only be used with fields that support ordering and can't be used with text fields. For example, date fields and version fields.

To see a field's supported operators, check the individual [field reference](#).

Examples

Find all issues with less than votes:

```
votes < 4
```

[^top of page](#)

LESS THAN EQUALS: <=

The <= operator is used to search for issues where the value of a specified field is less than or equal to a specified value.

i The operator can only be used with fields that support ordering and can't be used with text fields. For example, date fields and version fields.

To see a field's supported operators, check the individual [field reference](#).

Examples

- Find all issues with four or fewer votes:

```
votes <= 4
```

- Find all issues that haven't been updated in the last month (30 days):

```
updated <= "-4w 2d"
```

[^top of page](#)

IN

The `IN` operator is used to search for issues where the value of a specified field is one of multiple specified values. The values are specified as a comma-separated list, surrounded by parentheses.

i Using `IN` is equivalent to using multiple `EQUALS (=)` statements, but is shorter and more convenient. That is, `reporter IN (tom, jane, harry)` is the same as `reporter = "tom" OR reporter = "jane" OR reporter = "harry"`.

Examples

- Find all issues that were created by either `jsmith`, `jbrown`, or `jjones`:

```
reporter in (jsmith,jbrown,jjones)
```

- Find all issues where the Reporter or Assignee is either Jack or Jill:

```
reporter in (Jack,Jill) or assignee in (Jack,Jill)
```

- Find all issues in version 3.14 or version 4.2:

```
affectedVersion in ("3.14", "4.2")
```

[^top of page](#)

NOT IN

The `NOT IN` operator is used to search for issues where the value of a specified field isn't one of multiple specified values.

i Using `NOT IN` is equivalent to using multiple `NOT_EQUALS (!=)` statements, but is shorter and more convenient. That is, `reporter NOT IN (tom, jane, harry)` is the same as `reporter != "tom" AND reporter != "jane" AND reporter != "harry"`.

Also, the `NOT IN` operator won't match a field that has no value (a field is empty). For example, `assignee not in (jack, jill)` will only match issues that have an assignee and this assignee isn't "jack" or "jill".

To find issues that are assigned to someone other than "jack" or "jill" or are unassigned, you should type: `assignee not in (jack, jill) or assignee is empty`.

Examples

- Find all issues where the Assignee is someone other than Jack, Jill, or John:

```
assignee not in (Jack,Jill,John)
```

- Find all issues where the Assignee isn't Jack, Jill, or John:

```
assignee not in (Jack,Jill,John) or assignee is empty
```

- Find all issues where the fix version isn't A, B, C, or D:

```
FixVersion not in (A, B, C, D)
```

- Find all issues where the fix version isn't A, B, C, or D, or has not been specified:

```
FixVersion not in (A, B, C, D) or FixVersion is empty
```

[^top of page](#)

CONTAINS: ~

The ~ operator is used to search for issues where the value of a specified field matches a specified value: either an exact or fuzzy match. (see examples below).


Use it only with version and text [fields](#).

Text fields:


- Summary
- Description
- Environment
- Comments
- Custom fields that use the free text searcher, including custom fields of the following built-in custom field types:
 - Free text field (unlimited text)
 - Text field (<255 characters)
 - Read-only text field

Version fields:

- Affected version
- Fix version
- Custom fields that use the version picker

 The JQL field "text", as `intext ~ "some words"`, searches for an issue's summary, description, environment, comments, and all custom text fields.

If you have many text custom fields, you can improve performance of your queries by searching for specific fields. For example: `Summary ~ "some words" OR Description ~ "some words"`.

 When using the ~ operator, the value on the right side of the operator can be specified by using [Jira text-search syntax](#).

Examples

- Find all issues where the summary contains the word "win" or the simple derivatives of this word, such as "wins":

```
summary ~ win
```

- Note that for version fields, the ~ operator returns an exact match. For example, to find the version 9.0, you should use the following query:

```
fixVersion ~ "9.0"
```

- Find all issues where the summary contains a wild card match for the word "win":


```
summary ~ "win*"
```

- Find all issues where the summary contains the word "issue" and the word "collector":

```
summary ~ "issue collector"
```

- Find all issues where the summary contains the exact phrase "full screen". Also, see [Search syntax for text fields](#) for details on how to escape quotation marks and other special characters.

```
summary ~ "\"full screen\""
```

- Find all issues where the **Fix Version** field contains a wild card match for the version "9". For example, 9.1 or 9.0.1:

```
fixVersion ~ "9*"
```

- Find all issues where the **Fix Version** field contains "9". For example, 1.9:

```
fixVersion ~ "*9"
```

[^top of page](#)

DOES NOT CONTAIN: !~

The !~ operator is used to search for issues where the value of a specified field matches a specified value.

Use it only with version and text [fields](#).

Text fields:

- Summary
- Description
- Environment
- Comments
- Custom fields that use the free text searcher, including custom fields of the following built-in custom field types:
 - Free text field (unlimited text)
 - Text field (<255 characters)
 - Read-only text field

Version fields:

- Affected version
- Fix version
- Custom fields that use the Version Picker



The JQL field "text", as `intext !~ "some words"`, searches for an issue's summary, description, environment, comments, and all custom text fields.

If you have many text custom fields, you can improve performance of your queries by searching for specific fields. For example: `Summary !~ "some words" OR Description !~ "some words"`.



When using the !~ operator, the value on the right side of the operator can be specified by using [Jira text-search syntax](#).

Examples

- Find all issues where the summary doesn't contain the word "run" or the derivatives of this word, such as "running" or "ran":

```
summary !~ run
```

- Note that for version fields, the ~ operator returns an exact match. For example, to find issues where the fix version is not 9.0, you should use the following query:

```
fixVersion !~ "9.0"
```


- Find all issues where the **Fix Version** field doesn't contain any version from the 9.x line:

```
fixVersion !~ "9.*"
```

[^top of page](#)

IS

The `IS` operator can only be used with the `EMPTY` or `NULL` keywords. That is, it's used to search for issues where the specified field has no value.

 Not all fields are compatible with this operator. For more details, see the individual [field reference](#).

Examples

- Find all issues that have no fix version:

```
fixVersion is empty
```


or

```
fixVersion is null
```

[^top of page](#)

IS NOT

The `IS NOT` operator can only be used with the `EMPTY` or `NULL` keywords. That is, it's used to search for issues where a specified field has a value.

 Not all fields are compatible with this operator. For more details, see the individual [field reference](#).

Examples

- Find all issues that have one or more votes:

```
votes is not empty
```

or

```
votes is not null
```

[^top of page](#)

WAS

The `WAS` operator is used to find issues that currently have or previously had a specified value for a specified field.

In a search query, with this operator, you can use the following:

- `AFTER "date"`
- `BEFORE "date"`
- `BY "username"`
- `DURING ("date1", "date2")`
- `ON "date"`

The `WAS` operator will match the value name (for example, "Resolved") that was configured in your system at the time when the field was changed.

The operator will also match the value ID associated with the value name. For example, it will match "4" as well as "Resolved".



The operator can be used only with the following [fields](#): Assignee, Fix Version, Priority, Reporter, Resolution, and Status.

Examples

- Find issues that currently have or previously had the status "In Progress":

```
status WAS "In Progress"
```

- Find issues that were resolved by Joe Smith before February 20:

```
status WAS "Resolved" BY jsmith BEFORE "2011/02/20"
```

- Find issues that were resolved by Joe Smith during 2010:

```
status WAS "Resolved" BY jsmith DURING ("2010/01/01", "2011/01/01")
```

[^top of page](#)

WAS IN

The `WAS IN` operator is used to find issues that currently have or previously had any of multiple specified values for a specified field. The values are specified as a comma-separated list, surrounded by parentheses.

In a search query, with this operator, you can use the following:

- `AFTER "date"`
- `BEFORE "date"`
- `BY "username"`
- `DURING ("date1", "date2")`
- `ON "date"`

The `WAS IN` operator will match the value name (for example, "Resolved") that was configured in your system at the time when the field was changed.

The operator will also match the value ID associated with the value name. For example, it will match "4" as well as "Resolved".

i Using `WAS IN` is equivalent to using multiple `WAS` statements, but is shorter and more convenient. That is, `status WAS IN ("Resolved", "Closed")` is the same as `status WAS "Resolved" OR status WAS "Closed"`.

The operator can be used only with the following [fields](#): Assignee, Fix Version, Priority, Reporter, Resolution, and Status.

Examples

- Find all issues that currently have or previously had the status "Resolved" or "In Progress":

```
status WAS IN ("Resolved", "In Progress")
```

[^top of page](#)

WAS NOT IN

The `WAS NOT IN` operator is used to search for issues where the value of the specified field has never been one of multiple specified values.

In a search query, with this operator, you can use the following:

- AFTER "date"
- BEFORE "date"
- BY "username"
- DURING ("date1", "date2")
- ON "date"

i Using `WAS NOT IN` is equivalent to using multiple `WAS NOT` statements, but is shorter and more convenient. That is, `status WAS NOT IN ("Resolved", "In Progress")` is the same as `status WAS NOT "Resolved" AND status WAS NOT "In Progress"`.

The operator can be used only with the following [fields](#): Assignee, Fix Version, Priority, Reporter, Resolution, and Status.

Examples

- Find issues that have never had the status "Resolved" or "In Progress":

```
status WAS NOT IN ("Resolved", "In Progress")
```

- Find issues that didn't have the status "Resolved" or "In Progress" before February 20:

```
status WAS NOT IN ("Resolved", "In Progress") BEFORE "2011/02/20"
```

[^top of page](#)

WAS NOT

The `WAS NOT` operator is used to find issues that have never had a specified value for a specified field.

In a search query, with this operator, you can use the following:

- AFTER "date"
- BEFORE "date"
- BY "username"
- DURING ("date1", "date2")

- ON "date"

The `WAS NOT` operator will match the value name (for example, "Resolved") that was configured in your system at the time when the field was changed.

The operator will also match the value ID associated with the value name. For example, it will match "4" as well as "Resolved".

 The operator can be used only with the following [fields](#): Assignee, Fix Version, Priority, Reporter, Resolution, and Status.

Examples

- Find issues that don't have and have never had the status "In Progress":

```
status WAS NOT "In Progress"
```

- Find issues that didn't have the status "In Progress" before February 20:

```
status WAS NOT "In Progress" BEFORE "2011/02/20"
```

[^top of page](#)

CHANGED

The `CHANGED` operator is used to find issues where the value of a specified field was changed.

In a search query, with this operator, you can use the following:

- AFTER "date"
- BEFORE "date"
- BY "username"
- DURING ("date1", "date2")
- ON "date"
- FROM "oldvalue"
- TO "newvalue"

 The operator can be used only with the following [fields](#): Assignee, Fix Version, Priority, Reporter, Resolution, and Status.

Examples

- Find issues where the Assignee was changed:

```
assignee CHANGED
```

- Find issues where the status was changed from "In Progress" back to "Open":

```
status CHANGED FROM "In Progress" TO "Open"
```

- Find issues where the priority was changed by the user `freddo` after the start and before the end of the current week:

```
priority CHANGED BY freddo BEFORE endOfWeek() AFTER startOfWeek()
```

[^top of page](#)

Advanced searching - functions reference

This page describes information about functions that are used for advanced searching.

A function in JQL appears as a word followed by parentheses, which may contain one or more explicit values or Jira [system](#) fields. In a clause, a function is preceded by an [operator](#), which in turn is preceded by [a field](#). A function performs a calculation on either specific Jira data or the function's content in parentheses, such that only true results are retrieved by the function, and then again by the clause in which the function is used.

Some Jira apps can add additional functions to the advanced issue search. For example, the `ScriptRunnerforJIRA` app extends JQL with such functions `asmyProjects()` and `projectmatch()`.

i Unless specified in the search query, note that JQL searches don't return empty fields in results. To include empty fields (e.g. unassigned issues) when searching for issues that are not assigned to the current user, you would enter `(assignee != currentUser() OR assignee is EMPTY)` to include unassigned issues in the list of results.

approved()

Only applicable if Jira Service Management is installed and licensed.

Search for issues that required approval and have a final decision of approved.

Syntax	<code>approved()</code>
Supported fields	Custom fields of type Approval
Supported operators	=
Unsupported operators	<code>~, !=, !~, >, >=, <, <=</code> <code>IS, IS NOT, IN, NOT IN, WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED</code>
Examples	<ul style="list-style-type: none">Find all issues that are approved: <code>approval = approved()</code>

[^ top of page](#)

On this page:

- [approved\(\)](#)
- [approver\(\)](#)
- [breached\(\)](#)
- [cascadeOption\(\)](#)
- [closedSprints\(\)](#)
- [completed\(\)](#)
- [componentsLeadByUser\(\)](#)
- [currentLogin\(\)](#)
- [currentUser\(\)](#)
- [earliestUnreleasedVersion\(\)](#)
- [elapsed\(\)](#)
- [endOfDay\(\)](#)
- [endOfMonth\(\)](#)
- [endOfWeek\(\)](#)
- [endOfYear\(\)](#)
- [everbreached\(\)](#)
- [futureSprints\(\)](#)
- [issueHistory\(\)](#)
- [issuesWithRemoteLinksByGlobalId\(\)](#)
- [lastLogin\(\)](#)
- [latestReleasedVersion\(\)](#)
- [linkedIssues\(\)](#)
- [membersOf\(\)](#)
- [myApproval\(\)](#)
- [myPending\(\)](#)
- [now\(\)](#)
- [openSprints\(\)](#)
- [paused\(\)](#)
- [pending\(\)](#)
- [pendingBy\(\)](#)
- [projectsLeadByUser\(\)](#)
- [projectsWhereUserHasPermission\(\)](#)
- [projectsWhereUserHasRole\(\)](#)
- [releasedVersions\(\)](#)
- [remaining\(\)](#)
- [running\(\)](#)
- [standardIssueTypes\(\)](#)
- [startOfDay\(\)](#)
- [startOfMonth\(\)](#)
- [startOfWeek\(\)](#)
- [startOfYear\(\)](#)
- [subtaskIssueTypes\(\)](#)
- [unreleasedVersions\(\)](#)
- [updatedBy\(\)](#)
- [votedIssues\(\)](#)
- [watchedIssues\(\)](#)
- [withinCalendarHours\(\)](#)

approver()

Only applicable if Jira Service Management is installed and licensed.

Search for issues that require or required approval by one or more of the listed users. This uses an OR operator, and you must specify the usernames.

Syntax	<code>approver(user, user)</code>
Supported fields	Custom fields of type Approval
Supported operators	=
Unsupported operators	~, !=, !~, >, >=, <, <= IS, IS NOT, IN, NOT IN, WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED
Examples	<ul style="list-style-type: none"> Find issues that require or required approval by John Smith: <code>approval = approver(jsmith)</code> Find issues that require or required approval by John Smith or Sarah Khan: <code>approval = approver(jsmith, skhan)</code>

[^ top of page](#)

breached()

Only applicable if Jira Service Management is installed and licensed.

Returns issues that whose most recent has missed its goal.

Syntax	<code>breached()</code>
Supported fields	
Supported operators	=, !=
Unsupported operators	~, !~, >, >=, <, <=, IS, IS NOT, WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED
Examples	<ul style="list-style-type: none"> Find issues where Time to First Response was breached: <code>"Time to First Response" = breached()</code>

[^ top of page](#)

cascadeOption()

Search for issues that match the selected values of a "cascading select" custom field.

The `parentOption` parameter matches against the first tier of options in the cascading select field. The `childOption` parameter matches against the second tier of options in the cascading select field, and is optional.


The keyword "none" can be used to search for issues where either or both of the options have no value.

Syntax	<code>cascadeOption(parentOption)</code> <code>cascadeOption(parentOption,childOption)</code>
Supported fields	Custom fields of type "Cascading Select"
Supported operators	IN , NOT IN
Unsupported operators	= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find issues where a custom field ("Location") has the value "USA" for the first tier and "New York" for the second tier: <code>location in cascadeOption("USA","New York")</code> Find issues where a custom field ("Location") has the value "USA" for the first tier and any value (or no value) for the second tier: <code>location in cascadeOption("USA")</code> Find issues where a custom field ("Location") has the value "USA" for the first tier and no value for the second tier: <code>location in cascadeOption("USA",none)</code> Find issues where a custom field ("Location") has no value for the first tier and no value for the second tier: <code>location in cascadeOption(none)</code> Find issues where a custom field ("Referrer") has the value "none" for the first tier and "none" for the second tier: <code>referrer in cascadeOption("\\"none\\"","\"none\\")</code> Find issues where a custom field ("Referrer") has the value "none" for the first tier and no value for the second tier: <code>referrer in cascadeOption("\\"none\\"",none)</code>

[^ top of page](#)

closedSprints()

Search for issues that are assigned to a completedSprint.

 It's possible for an issue to belong to both a completed Sprint(s) and an incomplete Sprint(s). See also [openSprints\(\)](#).

Syntax	<code>closedSprints()</code>
Supported fields	Sprint
Supported operators	IN , NOT IN
Unsupported operators	= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find all issues that are assigned to a completedsprint: <code>sprint in closedSprints()</code>

[^ top of page](#)

completed()

Only applicable if Jira Service Management is installed and licensed.

Returns issues that have an that has completed at least one cycle.

Syntax	<code>completed()</code>
Supported fields	
Supported operators	<code>= , !=</code>
Unsupported operators	<code>= , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Examples	<ul style="list-style-type: none"> Find issues where Time to First Response has completed at least one cycle: <code>"Time to First Response" = completed()</code>

[^ top of page](#)

componentsLeadByUser()

Find issues in components that are led by a specific user. You can optionally specify a user, or if the user is omitted, the current user (i.e. you) will be used.

 If you are not logged in to Jira, a user must be specified.

Syntax	<code>componentsLeadByUser()</code> <code>componentsLeadByUser(username)</code>
Supported fields	Component
Supported operators	<code>IN , NOT IN</code>
Unsupported operators	<code>= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Examples	<ul style="list-style-type: none"> Find open issues in components that are led by you: <code>component in componentsLeadByUser() AND status = Open</code> Find open issues in components that are led by Bill: <code>component in componentsLeadByUser(bill) AND status = Open</code>

[^ top of page](#)

currentLogin()


Perform searches based on the time at which the current user's session began. See also [lastLogin\(\)](#).

Syntax	<code>currentLogin()</code>
Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time
Supported operators	<code>= , != , > , >= , < , <=</code> <code>WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*</code> <i>*Only in predicate</i>
Unsupported operators	<code>~ , !~ , IS , IS NOT , IN , NOT IN</code>
Examples	<ul style="list-style-type: none"> Find issues that have been created during my current session: <code>created > currentLogin()</code>

[^ top of page](#)

currentUser()

Perform searches based on the currently logged-in user.

 This function can only be used by logged-in users. If you are creating a saved filter that you expect to be used by anonymous users, don't use this function.

Syntax	<code>currentUser()</code>
Supported fields	Assignee, Reporter, Voter, Watcher, custom fields of type User
Supported operators	<code>= , !=</code>
Unsupported operators	<code>~ , !~ , > , >= , < , <= , IS , IS NOT , IN , NOT IN , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Examples	<ul style="list-style-type: none"> Find issues that are assigned to me: <code>assignee = currentUser()</code> Find issues that were reported to me but are not assigned to me: <code>reporter = currentUser() AND (assignee != currentUser() OR assignee is EMPTY)</code>

[^ top of page](#)

earliestUnreleasedVersion()

Perform searches based on the earliest unreleased version (i.e. next version that is due to be released) of a specified project. See also [unreleasedVersions](#).

Consider that the "earliest" is determined by the ordering assigned to the versions, not by actual Version Due Dates.

Syntax	<code>earliestUnreleasedVersion(project)</code>
---------------	---

Supported fields	AffectedVersion, FixVersion, custom fields of type Version
Supported operators	IN , NOT IN
Unsupported operators	= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find issues whose FixVersion is the earliest unreleased version of the ABC project: fixVersion = earliestUnreleasedVersion(ABC) Find issues that relate to the earliest unreleased version of the ABC project: affectedVersion = earliestUnreleasedVersion(ABC) or fixVersion = earliestUnreleasedVersion(ABC)

[^ top of page](#)

elapsed()

Only applicable if Jira Service Management is installed and licensed.

Returns issues whose clock is at a certain point relative to a cycle's start event.

Syntax	elapsed()
Supported fields	
Supported operators	= , != , > , >= , < , <=
Unsupported operators	~ , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find issues that have been waiting for a first response for more than 1 hour: "Time to First Response" > elapsed("1h")

[^ top of page](#)

endOfDay()

Perform searches based on the end of the current day. See also [endOfWeek\(\)](#), [endOfMonth\(\)](#), and [endOfYear\(\)](#), [startOfDay\(\)](#), [startOfWeek\(\)](#), [startOfMonth\(\)](#), and [startOfYear\(\)](#).

Syntax	<p>endOfDay()</p> <p>endOfDay("inc")</p> <p>where <i>inc</i> is an optional increment of (+/-)nn(y M w d h m). If the time unit qualifier is omitted, it defaults to the natural period of the function, e.g. endOfDay("+1") is the same as endOfDay("+1d"). If the plus/minus (+/-) sign is omitted, plus is assumed.</p>
Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time

Supported operators	<p>= , != , > , >= , < , <=</p> <p>WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*</p> <p><i>*Only in predicate</i></p>
Unsupported operators	<p>~ , !~ , IS , IS NOT , IN , NOT IN</p>
Examples	<ul style="list-style-type: none"> Find issues due by the end of today: due < endOfDay() Find issues due by the end of tomorrow: due < endOfDay(" +1 ")

[^ top of page](#)

endOfMonth()

Perform searches based on the end of the current month. See also [endOfDay\(\)](#), [endOfWeek\(\)](#), [endOfYear\(\)](#), [startOfDay\(\)](#), [startOfWeek\(\)](#), [startOfMonth\(\)](#), and [startOfYear\(\)](#).

Syntax	<p>endOfMonth()</p> <p>endOfMonth("inc")</p> <p><i>where inc is an optional increment of (+/-)nn(y M w d h m) . If the time unit qualifier is omitted, it defaults to the natural period of the function, e.g. endOfMonth("+1") is the same as endOfMonth("+1M"). If the plus/minus (+/-) sign is omitted, plus is assumed.</i></p>
Supported fields	<p>Created, Due, Resolved, Updated, custom fields of type Date/Time</p>
Supported operators	<p>= , != , > , >= , < , <=</p> <p>WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*</p> <p><i>*Only in predicate</i></p>
Unsupported operators	<p>~ , !~ , IS , IS NOT , IN , NOT IN</p>
Examples	<ul style="list-style-type: none"> Find issues due by the end of this month: due <= endOfMonth() Find issues due by the end of next month: due <= endOfMonth(" +1 ") Find issues due by the 15th of next month: due <= endOfMonth(" +15d ")

[^ top of page](#)

endOfWeek()

Perform searches based on the end of the current week. See also [endOfDay\(\)](#), [endOfMonth\(\)](#), [endOfYear\(\)](#), [startOfDay\(\)](#), [startOfWeek\(\)](#), [startOfMonth\(\)](#), and [startOfYear\(\)](#).

For the `endOfWeek()` function, the result depends upon your locale. For example, in Europe, the first day of the week is generally considered to be Monday, while in the USA, it is considered to be Sunday.

Syntax	<pre>endOfWeek() endOfWeek("inc")</pre> <p>where <i>inc</i> is an optional increment of (+/-)nn(y M w d h m). If the time unit qualifier is omitted, it defaults to the natural period of the function, e.g. <code>endOfWeek("+1")</code> is the same as <code>endOfWeek("+1w")</code>. If the plus/minus (+/-) sign is omitted, plus is assumed.</p>
Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time
Supported operators	<pre>= , != , > , >= , < , <= WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*</pre> <p><i>*Only in predicate</i></p>
Unsupported operators	~ , !~ , IS , IS NOT , IN , NOT IN
Examples	<ul style="list-style-type: none"> Find issues due by the end of this week: <code>due < endOfWeek()</code> Find issues due by the end of next week: <code>due < endOfWeek("+1")</code>

[^ top of page](#)

endOfYear()

Perform searches based on the end of the current year. See also [startOfDay\(\)](#), [startOfWeek\(\)](#), [startOfMonth\(\)](#), [endOfDay\(\)](#), [endOfWeek\(\)](#), [endOfMonth\(\)](#), and [endOfYear\(\)](#).

Syntax	<pre>endOfYear() endOfYear("inc")</pre> <p>where <i>inc</i> is an optional increment of (+/-)nn(y M w d h m). If the time unit qualifier is omitted, it defaults to the natural period of the function, e.g. <code>endOfYear("+1")</code> is the same as <code>endOfYear("+1y")</code>. If the plus/minus (+/-) sign is omitted, plus is assumed.</p>
Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time
Supported operators	<pre>= , != , > , >= , < , <= WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*</pre> <p><i>*Only in predicate</i></p>
Unsupported operators	~ , !~ , IS , IS NOT , IN , NOT IN
Examples	<ul style="list-style-type: none"> Find issues due by the end of this year: <code>due < endOfYear()</code> Find issues due by the end of March next year: <code>due < endOfYear("+3M")</code>

[^ top of page](#)

everbreached()

Only applicable if Jira Service Management is installed and licensed.

Returns issues that have missed one of their goals.

Syntax	<code>elapsed()</code>
Supported fields	
Supported operators	<code>= , !=</code>
Unsupported operators	<code>~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Examples	<ul style="list-style-type: none"> Find issues have missed their goal for Time to First Response: <code>"Time to First Response" = everbreached()</code>

[^ top of page](#)

futureSprints()

Search for issues that are assigned to a sprint that hasn't been started yet.

It is possible for an issue to belong to both completed and incomplete sprints.

Syntax	<code>futureSprints()</code>
Supported fields	Sprint
Supported operators	<code>IN , NOT IN</code>
Unsupported operators	<code>= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Examples	<ul style="list-style-type: none"> Find all issues that are assigned to a sprint that hasn't been started yet: <code>sprint in futureSprints()</code>

[^ top of page](#)

issueHistory()

Find issues that you have recently viewed, i.e. issues that are in the **Recent Issues** section of the **Issues** dropdown menu.

 Note that:

- `issueHistory()` returns up to 60 issues, whereas the **Recent Issues** drop-down returns only 5.
- if you are not logged in to Jira, only issues from your current browser session will be included.
- issues older than 90 days are deleted daily by the scheduled job.

Syntax	<code>issueHistory()</code>
Supported fields	Issue
Supported operators	IN , NOT IN
Unsupported operators	= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find issues which I have recently viewed, that are assigned to me: <code>issue in issueHistory() AND assignee = currentUser()</code>

[^ top of page](#)

issuesWithRemoteLinksByGlobalId()

Perform searches based on issues that are associated with remote links that have any of the specified global IDs.

 This function accepts 1 to 100 globalIds. Specifying 0 or more than 100 globalIds will result in errors.

Syntax	<code>issuesWithRemoteLinksByGlobalId()</code>
Supported fields	Issue
Supported operators	IN , NOT IN
Unsupported operators	= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find issues that are linked to remote links that have globalId "abc": <code>issue in issuesWithRemoteLinksByGlobalId(abc)</code> Find issues that are linked to remote links that have either globalId "abc" or "def": <code>issue in issuesWithRemoteLinksByGlobalId(abc, def)</code>

[^ top of page](#)

lastLogin()

Perform searches based on the time at which the current user's previous session began. See also [currentLogin\(\)](#).

Syntax	<code>lastLogin()</code>
Supported fields	Created. Due, Resolved, Updated, custom fields of type Date/Time

Supported operators	<code>= , != , > , >= , < , <=</code> <code>WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*</code> <i>*Only in predicate</i>
Unsupported operators	<code>~ , !~ , IS , IS NOT , IN , NOT IN</code>
Examples	<ul style="list-style-type: none"> Find issues that have been created during my last session: <code>created > lastLogin()</code>

[^ top of page](#)

latestReleasedVersion()

Perform searches based on the latest released version (i.e. the most recent version that has been released) of a specified project. See also [releasedVersions\(\)](#).

Consider that the "latest" is determined by the ordering assigned to the versions, not by actual Version Due Dates.

Syntax	<code>latestReleasedVersion(project)</code>
Supported fields	AffectedVersion, FixVersion, custom fields of type Version
Supported operators	<code>= , !=</code>
Unsupported operators	<code>~ , !~ , > , >= , < , <=</code> <code>IS , IS NOT , IN , NOT IN , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Examples	<ul style="list-style-type: none"> Find issues whose FixVersion is the latest released version of the ABC project: <code>fixVersion = latestReleasedVersion(ABC)</code> Find issues that relate to the latest released version of the ABC project: <code>affectedVersion = latestReleasedVersion(ABC) or fixVersion = latestReleasedVersion(ABC)</code>

[^ top of page](#)

linkedIssues()

Perform searches based on issues that are linked to a specified issue. You can optionally restrict the search to links of a particular type.

Note that LinkType is case-sensitive.

Syntax	<code>linkedIssues(issueKey)</code> <code>linkedIssues(issueKey,linkType)</code>
Supported fields	Issue

Supported operators	IN , NOT IN
Unsupported operators	= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find issues that are linked to a particular issue: issue in linkedIssues(ABC-123) Find issues that are linked to a particular issue via a particular type of link: issue in linkedIssues(ABC-123,"is duplicated by")

[^ top of page](#)

membersOf()

Perform searches based on the members of a particular group.

Syntax	membersOf(Group)
Supported fields	Assignee, Reporter, Voter, Watcher, custom fields of type User
Supported operators	IN , NOT IN
Unsupported operators	= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find issues where the Assignee is a member of the group "Jira-administrators": assignee in membersOf("Jira-administrators") Search through multiple groups and a specific user: reporter in membersOf("Jira-administators") or reporter in membersOf("Jira-core-users") or reporter=jsmith Search for a particular group, but exclude a particular member or members: assignee in membersOf() and assignee not in ("John Smith","Jill Jones") Exclude members of a particular group: assignee not in membersOf()

[^ top of page](#)

myApproval()

Only applicable if Jira Service Management is installed and licensed.

Search for issues that require approval or have required approval by the current user.

Syntax	myApproval()
Supported fields	Custom fields of type Approval
Supported operators	=

Unsupported operators	<p>~, !=, !~ ,>, >=, <, <=</p> <p>IS , IS NOT ,IN , NOT IN ,WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</p>
Examples	<ul style="list-style-type: none"> Find all issues that require or have required my approval <code>approval = myApproval()</code>

[^ top of page](#)

myPending()

Only applicable if Jira Service Management is installed and licensed.

Search for issues that require approval by the current user.

Syntax	<code>myPending()</code>
Supported fields	Custom fields of type Approval
Supported operators	=
Unsupported operators	<p>~, !=, !~ ,>, >=, <, <=</p> <p>IS , IS NOT ,IN , NOT IN ,WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</p>
Examples	<ul style="list-style-type: none"> Find all issues that require my approval <code>approval = myPending()</code>

[^ top of page](#)

now()

Perform searches based on the current time.

Syntax	<code>now()</code>
Supported fields	Created. Due, Resolved, Updated, custom fields of type Date/Time
Supported operators	<p>= , != , > , >= , < , <=</p> <p>WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*</p> <p><i>*Only in predicate</i></p>
Unsupported operators	~ , !~ ,IS , IS NOT , IN , NOT IN
Examples	<ul style="list-style-type: none"> Find issues that are overdue: <code>duedate < now() and status not in (closed, resolved)</code>

[^ top of page](#)

openSprints()

Search for issues that are assigned to a Sprint that has not yet been completed.

i It's possible for an issue to belong to both a completed Sprint(s) and an incomplete Sprint(s). See also [closedSprints\(\)](#).

Syntax	<code>openSprints()</code>
Supported fields	Sprint
Supported operators	IN , NOT IN
Unsupported operators	= , != , ~ , !~ , > , >= , < , <= IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find all issues that are assigned to asprint that has not yet been completed: <code>sprint in openSprints()</code>

[^ top of page](#)

paused()

Only applicable if Jira Service Management is installed and licensed.

Returns issues that have an that is paused due to a condition.

To find issues that are paused because they are outside calendar hours, use [withincalendarhours\(\)](#).

Syntax	<code>paused()</code>
Supported fields	
Supported operators	= , !=
Unsupported operators	~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find issues where Time to First Response is paused: <code>"Time to First Response" = paused()</code>

[^ top of page](#)

pending()

Only applicable if Jira Service Management is installed and licensed.

Search for issues that require approval.

Syntax	<code>pending()</code>
---------------	------------------------

Supported fields	Custom fields of type Approval
Supported operators	=
Unsupported operators	~, !=, !~, >, >=, <, <= IS, IS NOT, IN, NOT IN, WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED
Examples	<ul style="list-style-type: none"> Find all issues that require approval: <code>approval = pending()</code>

[^ top of page](#)

pendingBy()

Only applicable if Jira Service Management is installed and licensed.

Search for issues that require approval by one or more of the listed users. This uses an OR operator, and you must specify the usernames.

Syntax	<code>pendingBy(user1,user2)</code>
Supported fields	Custom fields of type Approval
Supported operators	=
Unsupported operators	~, !=, !~, >, >=, <, <= IS, IS NOT, IN, NOT IN, WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED
Examples	<ul style="list-style-type: none"> Find issues that require approval by John Smith: <code>approval = pendingBy(jsmith)</code> Find issues that require by John Smith or Sarah Khan: <code>approval = pendingBy(jsmith,skhan)</code>

[^ top of page](#)

projectsLeadByUser()

Find issues in projects that are led by a specific user. You can optionally specify a user, or if the user is omitted, the current user will be used.

 If you are not logged in to Jira, a user must be specified.

Syntax	<code>projectsLeadByUser()</code> <code>projectsLeadByUser(username)</code>
Supported fields	Project
Supported operators	IN, NOT IN

Unsupported operators	<code>= , != , ~ , !~ , > , >= , < , <=</code> <code>IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Examples	<ul style="list-style-type: none"> Find open issues in projects that are led by you: <code>project in projectsLeadByUser() AND status = Open</code> Find open issues in projects that are led by Bill: <code>project in projectsLeadByUser(bill) AND status = Open</code>

[^ top of page](#)

projectsWhereUserHasPermission()

Find issues in projects where you have a specific permission. Note, this function operates at the project level. This means that if a permission (e.g. "Edit Issues") is granted to the reporter of issues in a project, then you may see some issues returned where you are not the reporter, and therefore don't have the permission specified. Also note, this function is only available if you are logged in to Jira.

Syntax	<code>projectsWhereUserHasPermission(permission)</code> For the <code>permission</code> parameter, you can specify any of the permissions described on .
Supported fields	Project
Supported operators	<code>IN , NOT IN</code>
Unsupported operators	<code>= , != , ~ , !~ , > , >= , < , <=</code> <code>IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Examples	<ul style="list-style-type: none"> Find open issues in projects where you have the "Resolve Issues" permission: <code>project in projectsWhereUserHasPermission("Resolve Issues") AND status = Open</code>

[^ top of page](#)

projectsWhereUserHasRole()

Find issues in projects where you have a specific role. Note, this function is only available if you are logged in to Jira.

Syntax	<code>projectsWhereUserHasRole(rolename)</code>
Supported fields	Project
Supported operators	<code>IN , NOT IN</code>
Unsupported operators	<code>= , != , ~ , !~ , > , >= , < , <=</code> <code>IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>

Examples	<ul style="list-style-type: none"> Find open issues in projects where you have the "Developers" role: <code>project in projectsWhereUserHasRole("Developers") AND status = Open</code>
-----------------	--

[^ top of page](#)

releasedVersions()

Perform searches based on the released versions (i.e. versions that your Jira administrator has released) of a specified project. You can also search on the released versions of all projects, by omitting the *project* parameter. See also [latestReleasedVersion\(\)](#).

Syntax	<pre>releasedVersions() releasedVersions(project)</pre>
Supported fields	AffectedVersion, FixVersion, custom fields of type Version
Supported operators	IN , NOT IN
Unsupported operators	= , != , ~ , !~ , > , >= , < , <= IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find issues whose FixVersion is a released version of the ABC project: <code>fixVersion in releasedVersions(ABC)</code> Find issues that relate to released versions of the ABC project: <code>(affectedVersion in releasedVersions(ABC)) or (fixVersion in releasedVersions(ABC))</code>

[^ top of page](#)

remaining()

Only applicable if Jira Service Management is installed and licensed.

Returns issues whose clock is at a certain point relative to the goal.

Syntax	<pre>remaining()</pre>
Supported fields	
Supported operators	= , != , > , >= , < , <=
Unsupported operators	~ , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find issues that will breach Time to Resolution in the next two hours: <code>"Time to Resolution" < remaining("2h")</code>

[^ top of page](#)

running()

Only applicable if Jira Service Management is installed and licensed.

Returns issues that have an that is running, regardless of the calendar.

To find issues that are running based on calendar hours, use [withincalendarhours\(\)](#).

Syntax	<code>running()</code>
Supported fields	
Supported operators	<code>= , !=</code>
Unsupported operators	<code>~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Examples	<ul style="list-style-type: none"> Find issues where Time to First Response is running: <code>"Time to First Response" = running()</code>

[^ top of page](#)

standardIssueTypes()

Perform searches based on "standard" Issue Types, that is, search for issues that are not sub-tasks. See also [subtaskIssueTypes\(\)](#).

Syntax	<code>standardIssueTypes()</code>
Supported fields	Type
Supported operators	<code>IN , NOT IN</code>
Unsupported operators	<code>= , != , ~ , !~ , > , >= , < , <= IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED</code>
Examples	<ul style="list-style-type: none"> Find issues that are not subtasks (i.e. issues whose Issue Type is a standard issue type, not a subtask issue type): <code>issuetype in standardIssueTypes()</code>

[^ top of page](#)

startOfDay()

Perform searches based on the start of the current day. See also [startOfWeek\(\)](#), [startOfMonth\(\)](#), [startOfYear\(\)](#), [endOfDay\(\)](#), [endOfWeek\(\)](#), [endOfMonth\(\)](#), and [endOfYear\(\)](#).

Syntax	<pre>startOfDay() startOfDay("inc")</pre> <p>where <i>inc</i> is an optional increment of (+/-)nn(y M w d h m). If the time unit qualifier is omitted, it defaults to the natural period of the function, e.g. <code>startOfDay("+1")</code> is the same as <code>startOfDay("+1d")</code>. If the plus/minus (+/-) sign is omitted, plus is assumed.</p>
Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time
Supported operators	<pre>= , != , > , >= , < , <= WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*</pre> <p><i>*Only in predicate</i></p>
Unsupported operators	~ , !~ , IS , IS NOT , IN , NOT IN
Examples	<ul style="list-style-type: none"> Find new issues created since the start of today: <code>created > startOfDay()</code> Find new issues created since the start of yesterday: <code>created > startOfDay("-1")</code> Find new issues created in the last three days: <code>created > startOfDay("-3d")</code>

[^ top of page](#)

startOfMonth()

Perform searches based on the start of the current month. See also [startOfDay\(\)](#), [startOfWeek\(\)](#), [startOfYear\(\)](#), [endOfDay\(\)](#), [endOfWeek\(\)](#), [endOfMonth\(\)](#), and [endOfYear\(\)](#).

Syntax	<pre>startOfMonth() startOfMonth("inc")</pre> <p>where <i>inc</i> is an optional increment of (+/-)nn(y M w d h m). If the time unit qualifier is omitted, it defaults to the natural period of the function, e.g. <code>startOfMonth("+1")</code> is the same as <code>startOfMonth("+1M")</code>. If the plus/minus (+/-) sign is omitted, plus is assumed.</p>
Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time
Supported operators	<pre>= , != , > , >= , < , <= WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*</pre> <p><i>*Only in predicate</i></p>
Unsupported operators	~ , !~ , IS , IS NOT , IN , NOT IN
Examples	<ul style="list-style-type: none"> Find new issues created since the start of this month: <code>created > startOfMonth()</code> Find new issues created since the start of last month: <code>created > startOfMonth("-1")</code> Find new issues created since the 15th of this month: <code>created > startOfMonth("+14d")</code>

[^ top of page](#)

startOfWeek()

Perform searches based on the start of the current week. See also [startOfDay\(\)](#), [startOfMonth\(\)](#), [startOfYear\(\)](#), [endOfDay\(\)](#), [endOfWeek\(\)](#), [endOfMonth\(\)](#), and [endOfYear\(\)](#).

For the startOfWeek() function, the result depends upon your locale. For example, in Europe, the first day of the week is generally considered to be Monday, while in the USA, it is considered to be Sunday.

Syntax	<pre>startOfWeek() startOfWeek("inc")</pre> <p>where <i>inc</i> is an optional increment of $(+/-)nn(y M w d h m)$. If the time unit qualifier is omitted, it defaults to the natural period of the function, e.g. <code>startOfWeek("+1")</code> is the same as <code>startOfWeek("+1w")</code>. If the plus/minus (+/-) sign is omitted, plus is assumed.</p>
Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time
Supported operators	<pre>= , != , > , >= , < , <= WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*</pre> <p><i>*Only in predicate</i></p>
Unsupported operators	<pre>~ , !~ , IS , IS NOT , IN , NOT IN</pre>
Examples	<ul style="list-style-type: none"> Find new issues since the start of this week: <code>created > startOfWeek()</code> Find new issues since the start of last week: <code>created > startOfWeek("-1")</code>

[^ top of page](#)

startOfYear()

Perform searches based on the start of the current year. See also [startOfDay\(\)](#), [startOfWeek\(\)](#), [startOfMonth\(\)](#), [endOfDay\(\)](#), [endOfWeek\(\)](#), [endOfMonth\(\)](#), and [endOfYear\(\)](#).

Syntax	<pre>startOfYear() startOfYear("inc")</pre> <p>where <i>inc</i> is an optional increment of $(+/-)nn(y M w d h m)$. If the time unit qualifier is omitted, it defaults to the natural period of the function, e.g. <code>endOfYear("+1")</code> is the same as <code>endOfYear("+1y")</code>. If the plus/minus (+/-) sign is omitted, plus is assumed.</p>
Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time
Supported operators	<pre>= , != , > , >= , < , <= WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*</pre> <p><i>*Only in predicate</i></p>
Unsupported operators	<pre>~ , !~ , IS , IS NOT , IN , NOT IN</pre>

Examples	<ul style="list-style-type: none"> Find new issues since the start of this year: <code>created > startOfYear()</code> Find new issues since the start of last year: <code>created > startOfYear("-1")</code>
-----------------	--

[^ top of page](#)

subtaskIssueTypes()

Perform searches based on issues that are sub-tasks. See also [standardIssueTypes\(\)](#).

Syntax	<code>subtaskIssueTypes()</code>
Supported fields	Type
Supported operators	IN , NOT IN
Unsupported operators	= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find issues that are subtasks (i.e. issues whose Issue Type is a subtask issue type): <code>issuetype in subtaskIssueTypes()</code>

[^ top of page](#)

unreleasedVersions()

Perform searches based on the unreleased versions (i.e. versions that your Jira administrator has not yet released) of a specified project. You can also search on the unreleased versions of all projects, by omitting the `project` parameter. See also [earliestUnreleasedVersion\(\)](#).

Syntax	<code>unreleasedVersions()</code> <code>unreleasedVersions(project)</code>
Supported fields	AffectedVersion, FixVersion, custom fields of type Version
Supported operators	IN , NOT IN
Unsupported operators	= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find issues whose FixVersion is an unreleased version of the ABC project: <code>fixVersion in unreleasedVersions(ABC)</code> Find issues that relate to unreleased versions of the ABC project: <code>affectedVersion in unreleasedVersions(ABC)</code>

[^ top of page](#)

updatedBy()

Search for issues that were updated by a specific user, optionally within the specified time range. An update in this case includes creating an issue, updating any of the issue's fields, creating or deleting a comment, or editing a comment (only the last edit).

For the time range, use one of the following formats:

"YYYY/MM/dd"

"YYYY-MM-dd"


Or use "w" (weeks), or "d" (days) to specify a date relative to the current time. Unlike some other functions, `updatedBy` doesn't support values smaller than a day, and will always round them up to 1 day.

Syntax	<pre>updatedBy(user) updatedBy(user, dateFrom) updatedBy(user, dateFrom, dateTo)</pre>
Supported fields	Issuekey, and its aliases (id, issue, key)
Supported operators	IN , NOT IN
Unsupported operators	= , ~ , != , !~ , > , >= , < , <= IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find issues that were updated by John Smith: <pre>issuekey IN updatedBy(jsmith)</pre> Find issues that were updated by John Smith within the last 8 days: <pre>issuekey IN updatedBy(jsmith, "-8d")</pre> Find issues updated between June and September 2018: <pre>issuekey IN updatedBy(jsmith, "2018/06/01", "2018/08/31")</pre> If you try to find issues updated in the last hour, like in the following example, the time will be rounded up to 1 day, as smaller values aren't supported: <pre>issuekey IN updatedBy(jsmith, "-1h")</pre>

[^ top of page](#)

votedIssues()

Perform searches based on issues for which you have voted. Also, see the [Voter](#) field.

 This function can only be used by logged-in users.


Syntax	<code>votedIssues()</code>
Supported fields	Issue
Supported operators	IN , NOT IN
Unsupported operators	= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED

Examples	<ul style="list-style-type: none"> Find issues that you have voted for: <code>issue in votedIssues()</code>
-----------------	--

[^ top of page](#)

watchedIssues()

Perform searches based on issues that you are watching. Also, see the [Watcher](#) field.

 This function can only be used by logged-in users.

Syntax	<code>watchedIssues()</code>
Supported fields	Issue
Supported operators	IN , NOT IN
Unsupported operators	= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find issues that you are watching: <code>issue in watchedIssues()</code>

[^ top of page](#)

withinCalendarHours()

Only applicable if Jira Service Management is installed and licensed.

Returns issues that have an that is running according to the calendar.

For example, say your project has two SLAs that count Time to First Response. Some issues with this use a 9am-1pm calendar, and others use a 9am-5pm calendar.If an agent starts work at 3pm, they probably want to work on issues from the 9am-5pm agreement first. They can use `withincalendarhours()` to find all the issues where Time to First Response is running at 3pm.

Syntax	<code>withinCalendarHours()</code>
Supported fields	
Supported operators	= , !=
Unsupported operators	~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED
Examples	<ul style="list-style-type: none"> Find issues where Time to First Response is within calendar hours: <code>"Time to First Response" = withinCalendarHours()</code>

[^ top of page](#)

Search syntax for text fields

This page provides information on the syntax for searching text fields, which can be done in the quick search, basic search, and advanced search.

Text searches can be done in the advanced search when the [CONTAINS\(~\) operator](#) is used, e.g. `summary~"windows"`. It can also be done in quick search and basic search when searching on supported fields.

Acknowledgments: Jira uses Apache Lucene for text indexing, which provides a rich query language. Much of the information on this page is derived from the [Query Parser Syntax](#) page of the Lucene documentation.

On this page:

- [Query terms](#)
- [Term modifiers](#)
- [Boosting a term: ^](#)
- [Boolean operators](#)
- [Grouping](#)
- [Special characters](#)
- [Reserved words](#)
- [Word stemming](#)
- [Limitations](#)
- [Next steps](#)

Query terms

A query is broken up into **terms** and **operators**. There are two types of terms: **Single Terms** and **Phrases**.

A **Single Term** is a single word, such as "test" or "hello".

A **Phrase** is a group of words surrounded by double quotes, such as "hello dolly".

Multiple terms can be combined together with Boolean operators to form a more complex query (see below). If you combine multiple terms without specifying any Boolean operators, they will be joined using AND operators.

Note: All query terms in Jira are not case sensitive.


Term modifiers

Jira supports modifying query terms to provide a wide range of searching options.

[Exact searches \(phrases\)](#) | [Wildcard searches: ? and *](#) | [Fuzzy searches: ~](#) | [Prefix and Suffix search](#) | [Proximity searches](#)

Exact searches (phrases)

To find exact matches for **phrases**, for example *Jira Software*, you need to enclose the whole phrase in quote-marks ("). Otherwise, the search will return all issues that contain both words in no particular order - this would include *Jira Software*, but also *Jira is the best software!*.

 If you're using advanced search, you need to additionally escape each of the quote-marks with a backslash (\). For details, see the examples below or find your field in [Advanced search - field reference](#).

Examples

- **Basic search:** Find all issues that contain the phrase *Jira Software*:

```
Just type "Jira Software" into the search field.
```

- **Advanced search:** Find all issues that contain the words *Jira* and *Software*, in no particular order.

```
text ~ "Jira Software"
```

- **Advanced search:** Find all issues that contain the phrase *Jira Software*.


```
text ~ "\"Jira Software\""
```

- **Advanced search:** Find all issues that contain the URL `https://atlassian.com`:

```
text ~ "\"https://atlassian.com\""
```

As you can see in the two preceding examples, the query contains two pairs of quote-marks. The external ones are needed to meet the JQL rules and are not related to your search query. The same pair of quote-marks would be automatically added by Jira in the basic search after running your search.

Using special characters to create phrases


In previous versions of Jira, you could use some special characters to combine **terms** into **phrases**, for example `Jira+Software` or `Jira/Software`. This is no longer the case, as the mechanism used for searching has changed and the special characters surrounding **terms** are ignored.

Wildcard searches: ? and *

Jira supports single and multiple character wildcard searches.

To perform a single character wildcard search, use the "?" symbol.

To perform a multiple character wildcard search, use the "*" symbol.

 Wildcard characters need to be enclosed in quote-marks, as they are reserved characters in advanced search. Use quotations, e.g. `summary ~ "cha?k and che*"`

The single character wildcard search looks for terms that match that with the single character replaced. For example, to search for "text" or "test", you can use the search:

```
te?t
```

Multiple character wildcard searches look for 0 or more characters. For example, to search for Windows, Win95, or WindowsNT, you can use the search:

```
win*
```

You can also use the wildcard searches in the middle of a term. For example, to search for Win95 or Windows95, you can use the search:

```
wi*95
```

Fuzzy searches: ~

Jira supports fuzzy searches. To do a fuzzy search, use the tilde, "~", symbol at the end of a single word term. For example, to search for a term similar in spelling to "roam", use the fuzzy search:

```
roam~
```

This search will find terms like foam and roams.

Note: Terms found by the fuzzy search will automatically get a boost factor of 0.2.

Prefix and Suffix search

Jira supports searching for parts of the words. To perform such search, include either a prefix or a suffix of the word or phrase you're looking for. For example to look for a MagicBox issue, you can use either of the two search patterns:

Prefix search

```
summary ~ "magic*"
```

Suffix search

```
summary ~ "*box"
```

Proximity searches

Jira supports finding words that are within a specific distance away. To do a proximity search, use the tilde, "~", symbol at the end of a phrase. For example, to search for "atlassian" and "Jira" within 10 words of each other in a document, use the search:

```
"atlassian Jira"~10
```

Boosting a term: ^

Jira provides the relevance level of matching documents based on the terms found. To boost a term, use the caret, "^", symbol with a boost factor (a number) at the end of the term you are searching. The higher the boost factor, the more relevant the term will be.

Boosting allows you to control the relevance of a document by boosting its term. For example, if you are searching for

```
atlassian Jira
```

and you want the term "atlassian" to be more relevant, boost it using the ^ symbol along with the boost factor next to the term. You would type:

```
atlassian^4 Jira
```

This will make documents with the term atlassian appear more relevant. You can also boost Phrase Terms, as in the example:

```
"atlassian Jira"^4 querying
```

By default, the boost factor is 1. Although, the boost factor must be positive, it can be less than 1 (i.e. 0.2).

Boolean operators

Boolean operators allow terms to be combined through logic operators. Jira supports AND, "+", OR, NOT and "-" as Boolean operators.

 Boolean operators must be ALL CAPS.

[AND](#) | [OR](#) | [Required term: +](#) | [NOT](#) | [Excluded term: -](#)

AND

The AND operator is the default conjunction operator. This means that if there is no Boolean operator between two terms, the AND operator is used. The AND operator matches documents where both terms exist anywhere in the text of a single document. This is equivalent to an intersection using sets. The symbol `&&` can be used in place of the word AND.

To search for documents that contain "atlassian Jira" and "issue tracking", use the query:

```
"atlassian Jira" AND "issue tracking"
```

OR

The OR operator links two terms, and finds a matching document if either of the terms exist in a document. This is equivalent to a union using sets. The symbol `||` can be used in place of the word OR.

To search for documents that contain either "atlassian Jira" or just "confluence", use the query:

```
"atlassian Jira" || confluence
```

or

```
"atlassian Jira" OR confluence
```

Required term: +

The "+" or required operator requires that the term after the "+" symbol exists somewhere in the field of a single document.

To search for documents that must contain "Jira" and may contain "atlassian", use the query:

```
+Jira atlassian
```

NOT

The NOT operator excludes documents that contain the term after NOT. This is equivalent to a difference using sets. The symbol `!` can be used in place of the word NOT.

To search for documents that contain "atlassian Jira" but not "japan", use the query:

```
"atlassian Jira" NOT "japan"
```

Note: The NOT operator cannot be used with just one term. For example, the following search will return no results:

```
NOT "atlassian Jira"
```



Usage of the **NOT** operator over multiple fields may return results that include the specified excluded term. This is due to the fact that the search query is executed over each field in turn, and the result set for each field is combined to form the final result set. Hence, an issue that matches the search query based on one field, but fails based on another field will be included in the search result set.

Excluded term: -

The "-" or prohibit operator excludes documents that contain the term after the "-" symbol.

To search for documents that contain "atlassian Jira" but not "japan", use the query:

```
"atlassian Jira" -japan
```

Grouping

Jira supports using parentheses to group clauses to form sub queries. This can be very useful if you want to control the boolean logic for a query.

To search for bugs and either atlassian or Jira, use the query:

```
bugs AND (atlassian OR Jira)
```

This eliminates any confusion and makes sure that bugs must exist, and either term atlassian or Jira may exist.

i Do not use the grouping character '(' at the start of a search query, as this will result in an error. For example, "(atlassian OR Jira) AND bugs" will not work.

Special characters

```
+ - & | ! ( ) { } [ ] ^ ~ * ? \ :
```

Special characters aren't stored in the index, which means you can't search for them. The index only keeps text and numbers, so searching for "\\[Jira Software\\]" and "Jira Software" will have the same effect. Escaped special characters ([]) won't be included in the search.

In previous Jira versions, you could use special characters to combine two separate terms into a phrase, for example "Jira+Software" or "Jira/Software". This doesn't apply to Jira 8.x. If you'd like to search for phrases, see [Exact searches \(phrases\)](#).

Reserved words

To keep the search index size and search performance optimal in Jira, the following English reserved words (also known as 'stop words') are ignored from the search index and hence, Jira's text search features:

"a", "and", "are", "as", "at", "be", "but", "by", "for", "if", "in", "into", "is", "it", "no", "not", "of", "on", "or", "such", "that", "the", "their", "then", "there", "these", "they", "this", "to", "was", "will", "with"

Be aware that this can sometimes lead to unexpected results. For example, suppose one issue contains the text phrase "VSX will crash" and another issue contains the phrase "VSX will not crash". A text search for "VSX will crash" will return both of these issues. This is because the words will and not are part of the reserved words list.

i Your Jira administrator can make Jira index these reserved words (so that Jira will find issues based on the presence of these words) by changing the **Indexing Language** to **Other** (under **Administration > System > General Configuration**).

Word stemming

Since Jira cannot search for issues containing parts of words (see [below](#)), word 'stemming' allows you to retrieve issues from a search based on the 'root' (or 'stem') forms of words instead of requiring an exact match with specific forms of these words. The number of issues retrieved from a search based on a stemmed word is typically larger, since any other issues containing words that are stemmed back to the same root will also be retrieved in the search results.

For example, if you search for issues using the query term 'customize' on the Summary field, Jira stems this word to its root form 'custom', and will retrieve all issues whose Summary field also contains any word that can be stemmed back to 'custom'. Hence, the following query:

```
summary ~ "customize"
```

will retrieve issues whose Summary field contains the following words:

- customized
- customizing
- customs
- customer
- etc.

ⓘ Please Note:

- Your Jira administrator can disable word stemming (so that Jira will find issues based on exact matches with words) by changing the **Indexing Language** to **Other** (under **Administration > System > General Configuration**).
- Word stemming applies to *all* Jira fields (as well as text fields).
- When Jira indexes its fields, any words that are 'stemmed' are stored in Jira's search index in root form only.

Limitations

Please note that the following limitations apply to Jira's search:

Whole words only

Jira cannot search for issues containing parts of words but on whole words only. The exception to this are words which are [stemmed](#).

This limitation can also be overcome using [fuzzy searches](#).

Next steps

Read the following related topics:

- [Searching for issues](#)
- [Quick searching](#)
- [Basic searching](#)
- [Advanced searching](#)

Saving your search as a filter

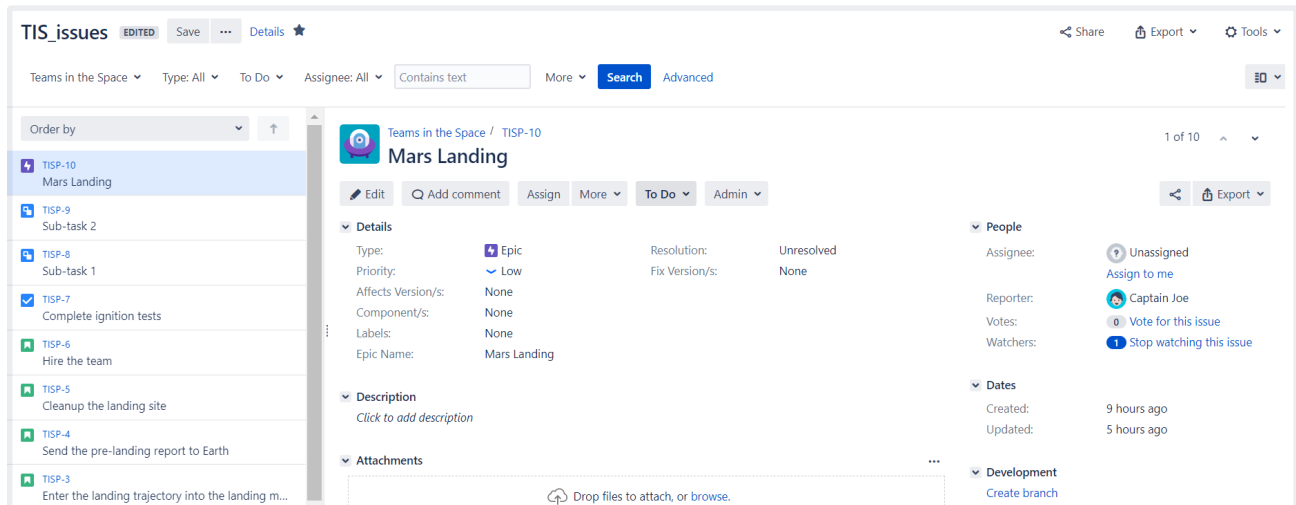
Jira's powerful [issue search](#) functionality is enhanced by the ability to save searches, called *filters* in Jira, for later use. You can do the following with Jira filters:

- Share and email search results with your colleagues, as well as people outside of your organization
- Create lists of [favorite filters](#)
- Have search results [emailed to you](#) according to your preferred schedule
- View and export the search results in various formats (RSS, Excel, etc)
- Display the search results in a [report format](#)
- Display the search results in a [dashboard gadget](#)

On this page:

- [Saving a search as a filter](#)
- [Running a filter](#)
- [Managing your existing filters](#)
- [Managing other user's shared filters](#)
- [Next steps](#)

Screenshot: Issue filter results in detail view



Saving a search as a filter

1. Define and run your search.
2. Click the **Save as** link above the search results. The **Save Filter** dialog is displayed.
3. Enter a name for the new filter and click **Submit**. Your filter is created.

Your new filter will be added to your favorite filters and shared, according to the sharing preference in your user profile. If you haven't specified a preference, then the global default will be applied, which is 'Private' unless changed by your Jira administrator.

Running a filter

1. Choose **Issues > Search for issues**.
2. Choose any filter from the list on the left:
 - System filter **My Open Issues, Reported by Me, Recently Viewed, All Issues**
 - Favorite filters (listed alphabetically)
 - **Find filters** lets you search for any filter that's been shared, which you can then subscribe to (adding it to your **Favorite Filters**).

3. After selecting a filter, the search results are displayed. The search criteria for the filter are also displayed and can be changed.

*Note, if you run the **Recently Viewed** system filter, this will switch you to the advanced search, as the basic search cannot represent the **ORDER BY** clause in this filter.*

Managing your existing filters

Click **Issues > Manage filters** to manage your filters.

Manage Filters			
Favorite My Popular Search	My Filters ⓘ Filters are issue searches that have been saved for re-use. This page shows all filters that you own.		
	Name	Shared With	Subscriptions
	★ Current sprint	• Private filter	None - Subscribe ⚙️
	☆ Due this week (TCYO)	• Project: Core	None - Subscribe ⚙️
	★ Epics	• Private filter	None - Subscribe ⚙️
	☆ Filter for EXP board	• Project: Space Exploration	None - Subscribe ⚙️
	★ Teams in Space	• Private filter	None - Subscribe ⚙️
	★ High-priority issues	• Private filter • Project: JIRA	None - Subscribe ⚙️
	★ Ignite docs	• Private filter	None - Subscribe ⚙️

The **Manage Filters** page allows you to view and configure filters that you have created, as well as work with filters that other users have shared with you. See the following topics for more information:

- [Searching for a filter](#)
- [Updating a filter](#)
- [Deleting a filter](#)
- [Cloning a filter](#)
- [Adding a filter as a favorite](#)
- [Sharing a filter](#)
- [Defining a filter-specific column order](#)
- [Subscribing to a filter](#)

Searching for a filter

You can find and run any filters that you have created or that have been shared by other users.

1. Click the **Search** tab on the 'Manage Filters' page.
2. Enter your search criteria and click **Search** to run the search.
3. Your search results are displayed on the same page. Click the name of any issue filter to run it.

*Tip: If the filter has been added as a favorite by many users, you may also be able locate it on the **Popular** tab of the **Manage Filters** page.*

Updating a filter

You can update the name, description, sharing, favorite of any filters that you created, or have permission to edit. If you want to edit a filter for which you only have the *view* permission, either [clone](#) (aka copy) the shared filter, or ask your Jira administrator to [change the filter's ownership](#).

Update the filter's details:

1. Click the **My** tab on the 'Manage Filters' page.
2. Locate the filter you wish to update, click the **cog icon > Edit**.
3. The **Edit Current Filter** page displays, where you can update the filter details as required.
4. Click **Save** to save your changes.

i If you have an editor role assigned and want to save changes to a filter, you **must** be a member of all groups that the filter is shared with. Otherwise, you will not be able to save the changes.

Update the filter's search criteria:

1. Click the **My** tab on the 'Manage Filters' page.
2. Locate the filter you want to update and run it.

3. Update the search criteria as desired, and rerun the query to ensure the update is valid. You will see the word *Edited* displayed next to your filter name.
4. Click **Save** to overwrite the current filter with the updated search criteria. If you want discard your changes instead, click the arrow next to the save button, and select **Discard changes**.

Deleting a filter

1. Click the **My** tab on the 'Manage Filters' page.
2. Locate the filter you wish to delete, click the **cog icon > Delete**.

Cloning a filter

You can clone any filter which is just a way of making a copy that you own that was either created by you or shared with you.

1. Locate the filter you wish to clone and run it.
2. Update the search criteria as desired. Click the arrow next to the **Save** button, and select **Save > Save as** to create a new filter from the existing filter.

Adding a filter as a favorite

Filters that you've created or that have been shared by others can be added to your favorite filters. Favorite filters are listed in the menu under **Issues > Filters**, and in the left panel of the issue navigator.

1. Locate the filter you wish to add as a favorite.
2. Click the star icon next to the filter name to add it to your favorites.

Sharing a filter

Filters that you have created or have permission to edit can be shared with other users, user groups, projects, and project roles. They can also be shared globally. You can choose whether you want to share the filter with the permission to edit, or only to view. Any filter that is shared is visible to users who have the 'Jira Administrators' global permission. See [Managing other users' shared filters](#) below.

1. Click the **My** tab on the 'Manage Filters' page.
2. Locate the filter you wish to share, click the **cog icon > Edit**.
3. Update the **Add Viewers** and **Add Editors** fields by selecting the user, group, project, or project role that you want to share the filter with, and clicking **Add**. Note that you can only share filters with groups /roles of which you are a member.

You need the Create Shared Object global permission to configure sharing for a filter. Contact your Jira administrator to obtain this permission.

4. Click **Save** to save your changes.

*Tip: You can also share your filter by running it, then clicking **Details > Edit Permissions**.*

Defining a filter-specific column order

You can add a defined column order to a saved filter, which displays the filter results according to the saved column order. Otherwise, the results are displayed according to your personal column order (if you have set this) or the system default.

*Tip: To display your configured column order in a filter subscription, select " for the 'Outgoing email format' in your **User Profile**. If you receive text emails from Jira, you won't be able to see your configured column order.*

To add a column layout to a saved filter:

1. Click the **My** tab on the 'Manage Filters' page.
2. Locate the filter you wish to update; click the filter's name to display the results. Be sure you are viewing the filter in the **List** view so that you see the columns.
3. Configure the column order as desired by clicking on the column name and dragging it to the new position. Your changes are saved and will be displayed the next time you view this filter.

To remove a filter's saved column layout:

1. Click the **My** tab on the 'Manage Filters' page.
2. Locate the filter you wish to update; click the filter's name to display the results. Be sure you are viewing the filter in the **List** view so that you see the columns.
3. Click the **Columns** option on the top right of the displayed columns, and select **Restore Defaults** in the displayed window.

Exporting column ordered issues

When the results of a saved filter are exported to Excel, the column order and choice of columns are those that were saved with the filter. Even if a user has configured a personal column order for the results on the screen, the **saved configuration** is used for the Excel export. To export using your own configuration, save a copy of the filter along with your configuration, and then export the results to Excel.

Subscribing to a filter

See [Working with search results](#).

Managing other user's shared filters

A **shared filter** is a filter whose creator has shared that filter with other users. Refer to [Sharing a filter](#) above for details. When a shared filter is created by a user, that user:

- Initially 'owns' the shared filter.
- Being the owner, can edit and modify the shared filter.

If you have the **Jira Administrators** global permission, you can manage shared filters that were created by other users. For instructions, see [Managing shared filters](#).

Next steps

Read the following related topics:

- [Searching for issues](#)
- [Basic searching](#)
- [Advanced searching](#)
- [Working with search results](#)

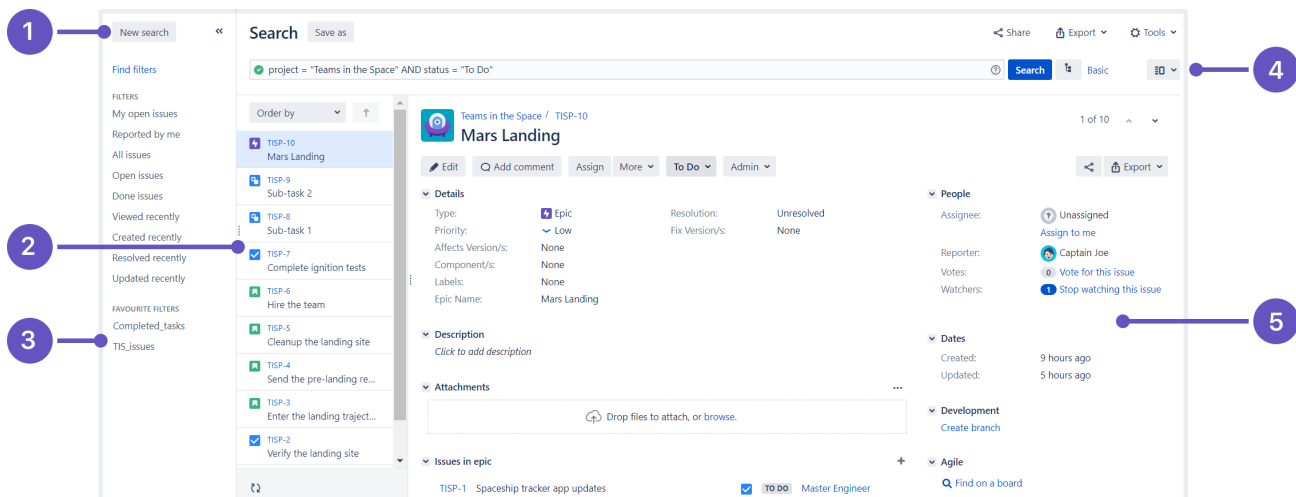
Working with search results

Once you have run a search, your search results will be displayed in the issue navigator. You may want to triage the entire list of issues or may be looking for just one. This page will show you what you can do with your search results, from changing what you see in the issue navigator to modifying the issues.

- On this page:**
- [Changing your view of the search results](#)
 - [Working with individual issues](#)
 - [Sharing your search results](#)
 - [Displaying your search results in Confluence](#)
 - [Displaying your search results as a chart](#)
 - [Exporting your search results](#)
 - [Printable views](#)
 - [Subscribing to your search results](#)
 - [Bulk modifying issues in your search results](#)
 - [Next steps](#)


The following screenshot provides an overview of the key features of the issue navigator.

Screenshot: Issue navigator (Detail view)



1. **Filter panel:** Click << to collapse the filter panel so you can have more space in the detail view.
2. **Issue:** Select an issue from this panel to see the details in the detail view window.
3. **Filters:** Select a filter to see all the matching issues in the panel to the immediate right.
4. **Views:** Click to switch between the detail view and list view.
5. **Detail view:** Check out all the details about the selected issue in this detail view.

Changing your view of the search results

<p>List view or Detail view</p>	<p>Click the Change view () dropdown to switch between List view and Detail view for your search results.</p> <ul style="list-style-type: none"> • List view: Shows your search results as a list of issues. This view is easiest to scan and is best when you only need to know a few details about each issue. • Detail view: Shows your search results as a list of issues, with the right panel showing the details of the currently selected issue. This view is best when you need more information about the individual issues, or you want to quickly edit issues as you go (via inline edit for certain fields).
<p>Change the sort order</p>	<p>Click the column name. If you click the same column name more than once, the sort order will switch between ascending and descending. Note:</p> <ul style="list-style-type: none"> • You cannot sort by the 'Images' column nor the sub-task aggregate columns (i.e. all columns beginning with "). • If you sort the search results for an advanced search, an 'ORDER BY' clause will be added/updated for your JQL query to reflect the order of issues in your search results.
<p>Columns show /hide and move</p>	<p>You can create different column configurations for yourself and for specific filters. To switch between different column configurations, click Columns and select one of the following tabs:</p> <ul style="list-style-type: none"> • My Defaults: This is your default column configuration for search results. • Filter: This is enabled if you are viewing the search results for a filter. It will override your default column configuration. • System (shows if you are a Jira administrator): This is the column configuration that applies to all users. It will be overridden by a user's default column configuration and filter-specific column configurations. <p>You can also modify any of these configurations. Make sure you have switched the desired configuration, then do the following:</p> <ul style="list-style-type: none"> • Show/hide columns: Click Columns, choose the desired columns, then click Done. • Move a column: Click the column name and drag it to the desired position. <p>If you cannot find a column, please make sure that you haven't run in to any of the following restrictions:</p> <ul style="list-style-type: none"> • You can only see columns for issue fields that have not been hidden and that you have permission to see. • It is possible to add any of the existing custom fields to the column list, as long as the fields are visible, and you have the right permissions. • Some custom fields, even if selected, do not appear in the Issue Navigator for all issues. For example, project-specific custom fields will be shown only if the filter has been restricted to that project only. Issue type custom fields will only appear if the filter has been restricted to that issue type.

Working with individual issues

You can action individual issues in your search results, directly from the issue navigator. Note that the list of issues will remain constant even if you change an issue, so that it doesn't meet the original search criteria. The advantage of this is that you have a constant set of search results that you can work from when triaging issues.

<p>View an issue</p>	<p>Click the key or summary of the issue.</p> <ul style="list-style-type: none"> • If you are in List view, you will be redirected to the issue (leaving the search results page). • If you are in Detail view, the issue details will display in the right panel.
-----------------------------	--

Action an issue	<p>To action an issue (e.g. edit it, transition it, log work on it, etc):</p> <ul style="list-style-type: none"> • If you are in List view, click the cog icon and select from the options. • If you are in Detail view, select the issue and action it the issue via the details panel. <p>You can also select an issue and action it via keyboard shortcuts in either views. <i>Tip: use the 'j' and 'k' keys to select the previous/next issue in the issue navigator.</i></p>
------------------------	--

Sharing your search results

Click **Share** in the issue navigator to email a link to a search result or shared filter.

- Recipients will receive an email with a link to the search result and the content of the **Note** field (if specified). The subject of the email will state that you (using your username) shared the issue.
- If you share the results of a filter, rather than an ad-hoc search, recipients will receive a link to the filter. Note, if the recipient does not have permission to view the filter, they will receive a link to the search results instead.

Displaying your search results in Confluence

If your Jira applications are connected to Confluence, you can display your search results on a Confluence page using the Jira issues macro. For instructions, see [Jira issues macro](#).

Displaying your search results as a chart

Click **Export > Dashboard charts**. Choose the desired chart from the dialog that is displayed, then click **Save to Dashboard**.

The chart will be added to your dashboard. For more information on what each chart shows, see [Reporting](#).

Exporting your search results

CSV	<p>Click Export > CSV (All fields) or Export > CSV (Current fields), and choose a delimiter to separate the values.</p> <p>The CSV file will contain a header row with every applicable issue field, comment, and attachment in your search result.</p> <ul style="list-style-type: none"> • CSV (All fields): this will include every issue field, comment and attachment. The header row may contain multiple values of "Comment" and/or "Attachment" if your issues have multiple comments and/or attachments. • CSV (Current fields): this will include only issue fields that are currently displayed. <p>Note, large exports (e.g. many hundreds of issues) are not recommended. You can change the number of issues that are exported, by changing the value of the tempMax parameter in the URL.</p> <p>If you're making a lot of exports, your Jira admin can disable the extra dialog that asks about delimiters. In this case, comma will be used as the default delimiter. Learn more</p>
------------	--

<p>HTML</p>	<p>Click Export > HTML (All fields) or Export > HTML (Current fields).</p> <p>The HTML file will contain a header row with a value for every applicable issue field in your search result.</p> <ul style="list-style-type: none"> • HTML (All fields): this will create an HTML file for every issue field (excluding comments). This will only show the custom fields that are available for all of the issues in the search results. For example, if a field is only available for one project and multiple projects are in the search results then that field will not appear in the HTML file. The same goes for fields that are only available for certain issue types. • HTML (Current fields): this will create an HTML file for the issue fields that are currently displayed. <p>Note, large exports (e.g. many hundreds of issues) are not recommended. You can change the number of issues that are exported, by changing the value of the tempMax parameter in the URL.</p>
<p>XML</p>	<p>Click Export > XML.</p> <p>You can use the URL of the XML view in a Confluence Jira issues macro. However, you can also use the JQL or the URL of the issue search, which are easier to get.</p> <p>To restrict which issue fields are returned in the XML export, specify the field parameter in your URL. For example, to include only the Issue key and Summary, add &field=key&field=summary to the URL. If the field parameter is not specified, the XML output will include all the issue fields. Otherwise, if one or more field parameters are specified, the XML output will contain only the Issue key plus your chosen field(s). See the "List of fields for field parameter" below.</p>
<p>Word</p>	<p>Click Export > Word.</p> <p>The export will include the Description, Comments, and all other issue data, not just the issue fields that are currently configured in your Issue Navigator. Note, large exports (e.g. hundreds of issues) are not recommended.</p>

List of fields for field parameter (XML exports):

Value	Sample XML output	Notes
<p>title</p>	<pre><title>[TEST-4] This is a test</title></pre>	
<p>link</p>	<pre><link>https://extranet.atlassian.com:443/Jira/browse/TEST-4</link></pre>	<p>This is a 'permalink' to the issue. For links between issues, see * issuelinks * (below).</p>
<p>project (or pid)</p>	<pre><project id="10330" key="TST">Test</project></pre>	
<p>description</p>	<pre><description>This is a detailed description of the issue.</description></pre>	
<p>environment</p>	<pre><environment>Sydney network</environment></pre>	
<p>key</p>	<pre><key id="22574">TEST-4</key></pre>	

summary	<code><summary>This is a test</summary></code>	
type (or issue type)	<code><type id="3" iconUrl="https://extranet.atlassian.com:443/Jira/images/icons/task.gif">Task</type></code>	
parent	<code><parent id="22620">TEST-5</parent></code>	Only relevant if the issue is a sub-task.
priority	<code><priority id="4" iconUrl="https://extranet.atlassian.com:443/Jira/images/icons/priority_minor.gif">Minor</priority></code>	
status	<code><status id="5" iconUrl="https://extranet.atlassian.com:443/Jira/images/icons/status_resolved.gif">Resolved</status></code>	
resolution	<code><resolution id="1">Fixed</resolution></code>	
labels	<code><labels> <label>focus</label> </labels></code>	
assignee	<code><assignee username="jsmith">John Smith</assignee></code>	
reporter	<code><assignee username="jsmith">John Smith</assignee></code>	
security	<code><security id="10021">Private</security></code>	Only relevant if a security level has been applied to the issue.
created	<code><created>Mon, 1 Sep 2008 17:30:03 -0500 (CDT)>/created></code>	
updated	<code><updated>Mon, 1 Sep 2008 17:30:03 -0500 (CDT)>/updated></code>	
resolved (or resolution date)	<code><resolved>Mon, 1 Sep 2008 17:30:03 -0500 (CDT)>/resolved></code>	

due (or due date)	<pre><due>Mon, 1 Sep 2008 17:30:03 -0500 (CDT) </due></pre>	
version (or versions)	<pre><version>2.4.7</version></pre>	
fixfor (or fix versions)	<pre><fixVersion>2.6</fixVersion></pre>	
component (or components)	<pre><component>Documentation</component></pre>	
votes	<pre><votes>1</votes></pre>	
comments (or comment)	<pre><comments> <comment id="39270" author="jsmith" created="Tue, 24 Feb 2009 16:45:02 -0600 (CST)">this looks familiar</comment> <comment id="39273" author="jbrown" created="Tue, 24 Feb 2009 16:48:16 -0600 (CST)">to me too</comment> </comments></pre>	
attachments (or attachment)	<pre><attachments> <attachment id="30318" name="Issue Navigator - Atlassian Jira-2.png" size="16161" author="yoz" created="Mon, 9 Feb 2009 13:32:58 -0600 (CST)"/> <attachment id="30323" name="Windows XP (with Firefox 3.0).jpg" size="5802" author="vbharara" created="Tue, 10 Feb 2009 00:30:11 -0600 (CST)"/> </attachments></pre>	Only available if your administrator has enabled attachments.
timeoriginal estimate	<pre><timeoriginalestimate seconds="600">10 minutes</timeoriginalestimate></pre>	Only available if your administrator has time tracking enabled.
timeestimate	<pre><timeestimate seconds="300">5 minutes</timeestimate></pre>	Only available if your administrator has time tracking enabled.
timespent	<pre><timespent seconds="300">5 minutes</timespent></pre>	Only available if your administrator has time tracking enabled.
aggregate time original estimate	<pre><aggregate time original estimate seconds="36000">10 hours</aggregate time original estimate></pre>	Aggregate time for the issue plus all of its sub-tasks. Only available if your administrator has time tracking enabled.

<p>aggregatetimeestimate</p>	<pre><aggregatetimerremainingestimate seconds="18000">5 hours</aggregatetimerremainingestimate></pre>	<p>Aggregate time for the issue plus all of its sub-tasks. Only available if your administrator has time tracking enabled.</p>
<p>aggregatetimespent</p>	<pre><aggregatetimespent seconds="18000">5 hours</aggregatetimespent></pre>	<p>Aggregate time for the issue plus all of its sub-tasks. Only available if your administrator has time tracking enabled.</p>
<p>timetracking</p>	<pre><timeoriginalestimate seconds="600">10 minutes</timeoriginalestimate> <timeestimate seconds="300">5 minutes</timeestimate> <timespent seconds="300">5 minutes</timespent> <aggregatetimeoriginalestimate seconds="36000">10 hours</aggregatetimeoriginalestimate> <aggregatetimerremainingestimate seconds="18000">5 hours</aggregatetimerremainingestimate> <aggregatetimespent seconds="18000">5 hours</aggregatetimespent></pre>	<p>This is a convenient shorthand way of specifying all of the above six time tracking fields. Only available if your administrator has time tracking enabled.</p>
<p>issuelinks</p>	<pre><issuelinks> <issuelinktype id="10020"> <name>Duplicate</name> <inwardlinks description="is duplicated by"> <issuelink> <issuekey id="22477">INTSYS-1009</issuekey> </issuelink> </inwardlinks> </issuelinktype> </issuelinks></pre>	
<p>subtasks (or subtask)</p>	<pre><subtasks> <subtask id="22623">TEST-8</subtask> </subtasks></pre>	
<p>customfield_xx xxx</p>	<pre><customfields> <customfield id="customfield_10112" key="com.atlassian.Jira.plugin.system.customfieldtypes:select"> <customfieldname>Department</customfieldname> <customfieldvalues> <customfieldvalue>Adminstration</customfieldvalue> </customfieldvalues> </customfield> </customfields></pre>	<p>"xxxxx" is the id of a given custom field e.g. this output is the result of specifying &field=customfield_10112</p>

allcustom	<pre> <customfields> <customfield id="customfield_10112" key="com.atlassian.Jira.plugin.system. customfieldtypes:select"> <customfieldname>Department< /customfieldname> <customfieldvalues> <customfieldvalue>Adminstration< /customfieldvalue> </customfieldvalues> </customfield> <customfield id="customfield_10111" key="com.atlassian.Jira.plugin.system. customfieldtypes:select"> <customfieldname>Expenditure Type</customfieldname> <customfieldvalues> <customfieldvalue>Operating< /customfieldvalue> </customfieldvalues> </customfield> </customfields> </pre>
------------------	---

Printable views

Printable	<p>Click Export > Printable.</p> <p>Creates a view of your search results in your browser that can be printed 'Landscape'. This view only contains issue Type, Key, Summary, Assignee, Reporter, Priority, Status, Resolution, Created date, Updated date, and Due date.</p>
Full content	<p>Click Export > Full content.</p> <p>Creates a view of your search results in your browser that can be printed. This view contains all issue fields, comments, and a list of attachments (there is no preview) for every issue returned by your search.</p>

Subscribing to your search results

A subscription provides you with a periodic notification for all issues returned by the search. If you want to be notified when a particular issue changes, you should watch the issue instead.

Email	<p>Your search must be saved as a filter, if you want to create an email subscription for it. You can create a subscription of any frequency for yourself and/or other users. Note, only the first 200 results of a filter are sent.</p> <ol style="list-style-type: none"> 1. Run the filter that you want to subscribe to, then click Details (next to filter name). 2. Fill in the 'Filter Subscription form' and click Subscribe. <p>More information:</p> <ul style="list-style-type: none"> • If you choose 'Advanced' for your Schedule, see this page for help on constructing Cron expressions. • You can choose to specify a group as a recipient, however you can only select a group that you are a member of: <ul style="list-style-type: none"> • You must have the 'Manage Group Filter Subscriptions' global permission. • Be aware that the emailed filter results will be specific to each recipient. For example, if the filter uses the <code>currentUser()</code> function, the search results will be evaluated with the recipient as the current user. This does not apply to distribution lists (group email aliases). • Be careful about sharing a subscription with a group with many members, as it can take a long time to generate the emails to be sent, since the search needs to be executed for each user (as per the previous point).
RSS	<p>Click Export > RSS (Issues) or Export > RSS (Comments). The URL of the page that shows can be used in your feed reader.</p> <p>Tips:</p> <ul style="list-style-type: none"> • You can change the number of issues that are returned, by changing the value of the tempMax parameter in the URL. • If you only want to receive current comments in an RSS feed, use the Date Updated field when doing a search. For example, to only receive comments created in the last week, add the Date Update field and set it to updated within the last 1 week. • You may need to log into your Jira applications to view restricted data in your feed. If so, you can add os_authType=basic to the feed URL (e.g. <code>http://mycompany.com/anypage?os_authType=basic</code>) to show a login dialog when viewing the feed.

Bulk modifying issues in your search results

Bulk operations let you action multiple issues at once. These actions include transitioning issues, deleting issues, moving issues, and watching/unwatching issues.

Click **Tools > Bulk Change: all <N> issue(s)** and follow the 'Bulk Operation' wizard.

For more information, see [Editing multiple issues at the same time](#).

Next steps

Read the following related topics:

- [Searching for issues](#)
- [Constructing cron expressions for a filter subscription](#)

Constructing cron expressions for a filter subscription

This page describes how to construct a cron expression. Cron expressions can be used when creating a subscription to a filter, as described in Working with search results.

A cron expression gives you more control over the frequency, compared to the default schedules. For example, you could define a cron expression to notify you at 8:15 am on the second Friday of every month.

Constructing a cron expression

A cron expression is a string of fields separated by spaces. The following table displays the fields of a cron expression, *in the order that they must be specified (from left to right)*:

	Second	Minute	Hour	Day-of-month	Month	Day-of-week	Year (optional)
Allowed values	0-59	0-59	0-23	1-31	1-12 or JAN-DEC	1-7 or SUN-SAT	1970-2099
Allowed special characters	, - * /	, - * /	, - * /	, - * / ? L W C	, - * /	, - * / ? L C #	, - * /

Note, cron expressions are not case-sensitive.

Here is an example:

```
0 15 8 ? JAN MON 2014
```

This literally translates to 0 second, 15 minute, 8 hour, any day of the month, January, 2014.

In plain English, this represents 8:15am onevery Monday during January of 2014. Note, the ? character means "no particular value". In this example, we've set the Day-of-month to no particular value. We don't need to specify it, as we've specified a Day-of-week value. Read more about special characters in the next section.

More examples of cron expressions are explained in the [Examples section](#) at the bottom of this page.

Special characters

Special character	Usage
,	Specifies a list of values. For example, in the Day-of-week field, 'MON,WED,FRI' means 'every Monday, Wednesday, and Friday'.
-	Specifies a range of values. For example, in the Day-of-week field, 'MON-FRI' means 'every Monday, Tuesday, Wednesday, Thursday and Friday'.
*	Specifies all possible values. For example, in the Hour field, '*' means 'every hour of the day'.
/	Specifies increments to the given value. For example, in the Minute field, '0/15' means 'every 15 minutes during the hour, starting at minute zero'.
?	Specifies no particular value. This is useful when you need to specify a value for one of the two fields Day-of-month or Day-of-week , but not the other.

L	Specifies the last possible value; this has different meanings depending on context. In the Day-of-week field, 'L' on its own means 'the last day of every week' (i.e. 'every Saturday'), or if used after another value, means 'the last xxx day of the month' (e.g. 'SATL' and '7L' both mean 'the last Saturday of the month'). In the Day-of-month field, 'L' on its own means 'the last day of the month', or 'LW' means 'the last weekday of the month'.
W	Specifies the weekday (Monday-Friday) nearest the given day of the month. For example, '1W' means 'the nearest weekday to the 1st of the month' (note that if the 1st is a Saturday, the email will be sent on the nearest weekday <i>within the same month</i> , i.e. on Monday 3rd). 'W' can only be used when the day-of-month is a single day, not a range or list of days.
#	Specifies the nth occurrence of a given day of the week. For example, 'TUES#2' (or '3#2') means 'the second Tuesday of the month'.

Examples

0 15 8 ? * *	Every day at 8:15 pm.
0 15 8 * * ?	Every day at 8:15 am.
0 * 14 * * ?	Every minute starting at 2:00 pm and ending at 2:59 pm, every day.
0 0/5 14 * * ?	Every 5 minutes starting at 2:00 pm and ending at 2:55 pm, every day.
0 0/5 14,18 * * ?	Every 5 minutes starting at 2:00 pm and ending at 2:55 pm, AND every 5 minutes starting at 6:00 pm and ending at 6:55 pm, every day.
0 0-5 14 * * ?	Every minute starting at 2:00 pm and ending at 2:05 pm, every day.
0 0/10 * * * ? *	Every 10 minutes, forever.
0 10,44 14 ? 3 WED	2:10 pm and 2:44 pm every Wednesday in the month of March.
0 15 8 ? * MON-FRI	8:15 am every Monday, Tuesday, Wednesday, Thursday, and Friday.
0 15 8 15 * ?	8:15 am on the 15th day of every month.
0 15 8 L * ?	8:15 am on the last day of every month.
0 15 8 LW * ?	8:15 am on the last weekday of every month.
0 15 8 ? * 6L	8:15 am on the last Friday of every month.
0 15 8 ? * 6#2	8:15 am on the second Friday of every month.
0 15 8 ? * 6#2 2007-2009	8:15 am on the second Friday of every month during the years 2007, 2008, and 2009.

Managing your user profile

You can manage your Jira settings (e.g. your password, email address, or the format in which you would like to receive email notifications) in your user profile. Your user profile also displays recent work in the Activity Stream, and contains useful shortcuts to issues you have been working on or reported.


To manage your user profile:


Choose **your user name** at top right of the screen, then choose **Profile**.

On this page:


- [Editing your user details](#)
- [Changing your avatar](#)
- [Choosing your homepage](#)
- [Managing email notifications](#)
- [Managing your user preferences](#)
- [Managing service desk preferences](#)
- [Managing your OAuth and login tokens](#)


Editing your user details

 If your instance is using an external user management system like Crowd, these options may not be available to you.

In the **Details** section on the **Summary** page, click the  at the top-right of the section to edit your display name, email address, and password. If your Jira administrator has configured the user directory with external password management, the **Change Password** link will not be available.

Changing your avatar

Select  or your current avatar to change the image that appears next to your name in Jira. If your administrator has [enabled Gravatar for user avatars](#), your Gravatar (i.e. the Gravatar associated with the email address in your user profile) will automatically be set as your user avatar. If Gravatar has been enabled, you will not be able to choose Jira-specific user avatars and vice versa. using [Gravatar.com](#). If Gravatar has been disabled, you can choose your user avatar from the ones pre-packaged with Jira or upload your own.

- 
- Your cropped image is resized to 48x48 pixels before it is saved as your new custom user avatar.
 - A separate 16x16 pixel version of your custom user avatar will be generated for use in comments.
 - Custom user avatars can only be selected by the user who uploaded them.

Choosing your homepage

Your Jira home page is the Jira page you are presented with immediately after you log in.


You can configure the following Jira pages as your Jira home page:

- The Dashboard
- The Issue Navigator
- Boards

1. Click on your **profile** icon at the top right of the screen.

2. Select the appropriate home page option within the **My Jira Home** section:
 - Dashboard
 - Issue Navigator
 - Boards (available if you're using Jira Software)
- Your page will be reloaded the Jira home page you selected.
3. (Optional) To verify that your Jira home page has been reset, log out and log back in to Jira again. You should be taken directly to the Jira home page you selected in the previous step.

Managing email notifications

In the **Preferences** section on the **Summary** page, click the **edit** icon  at the top-right of the section to open the **Updated User Preferences** dialog box. You can then manage the following:

- Change the **Email Type** to change the format (plain text or HTML) in which Jira sends its outgoing email notifications.
- In **My Changes**, Choose between making Jira send you email notifications about issue updates made by either both you and other people (**Notify me**) or other people only (i.e. **Do not notify me**).

Managing your user preferences

The global defaults for most of the user preferences below can be set by your Jira administrator; however, you can override these default settings by changing the following:

- The **Page Size**, or number of issues displayed on each Issue Navigator page
- Your preferred **language** from the drop-down list. If you don't see your preferred language in the list, see [Translating Jira](#) for more information.
- Your **time zone** specified in your profile doesn't match the time zone of the computer you are working on, Jira will ask if you want to update this selected time zone setting. All time fields in Jira will now be displayed in your preferred time zone.
- Choose the **Sharing** setting for when you create new filters and dashboards, which can be either shared with all other users (**Shared**) or restricted (**Unshared**).
- Choose to enable or disable Jira's keyboard shortcuts feature.
- Choose between allowing Jira to make you an **autowatcher** of any issue that you create or comment on.
- Choose how you want your external links to open. They can open in the existing or a new tab.

Managing service desk preferences

Service desk agents can enable or disable the **Pre-populated commenting** field by editing their user profiles. This setting can help save time by pre-filling conversation greeting text when agents comment on customer issues. When enabled, the text **Hi <Reporter_name>**, and **<Agent_name>** appears in the comment field and in the email notification sent to customers.

Managing your OAuth and login tokens

An OAuth access token is issued by Jira to give [gadgets](#) access to restricted data on an external, OAuth-compliant web application or website (also known as a "consumer"). Check out [Allowing OAuth access](#) for recommendations on when to issue or revoke OAuth access tokens.

If you are accessing your Jira applications in a public environment, you can clear your login tokens by clicking the **Clear all Tokens** link in the Details section of your Profile.

Allowing OAuth access

About OAuth access tokens

OAuth access tokens allow you to:


- Use a Jira gadget on an external, OAuth-compliant web application or website (also known as a 'consumer')
- Grant the gadget access to the same Jiradata that you can access.

Before you begin

Your Jira administrator must link your Jira instance and the consumer using an application link and OAuth. For example, if you want to add a Jira gadget to your Bamboo homepage, then your Jira administrator must first approve Bamboo as an OAuth consumer.

On this page:







- [About OAuth access tokens](#)
- [Before you begin](#)
- [Issuing OAuth access tokens](#)
- [Revoking OAuth access tokens](#)

 If you want to integrate with OAuth 2.0, see [Integrating with OAuth 2.0](#).

Issuing OAuth access tokens

To allow a gadget to access the same Jira data that you can, Jira issues it an OAuth access token. The OAuth token is unique to the gadget.

1. When you use a Jira gadget on a consumer (such as Bamboo) and this gadget requires access to your Jiradata, you will be prompted to log in to Jira if you have not already done so.
2. After you log in to Jira, you will be prompted with a **Request for Access** message.
3. To issue the OAuth token and grant the gadget access to your Jiradata, click **Allow**. The gadget can access your Jira data until you revoke the token.
4. To view tokens you have issued, go to your **Profile > Tools > View OAuth Access Tokens**:

Authorized Applications			
The following applications are using your account to access JIRA data			
 Bamboo on server-gdn-bamboo.internal.atlassian.com	Approved on Apr 26, 2016 at 7:09 AM	READ AND WRITE ACCESS	Revoke Access
 Confluence	Approved on Apr 22, 2016 at 5:53 AM	READ AND WRITE ACCESS	Revoke Access
 JDOG - JIRA Team Dogfood on jdog.jira-dev.com	Approved on Apr 27, 2016 at 12:43 AM	READ AND WRITE ACCESS	Revoke Access
 Stash on stash.atlassian.com	Approved on May 12, 2016 at 6:10 AM	READ AND WRITE ACCESS	Revoke Access
 Atlassian JIRA Extranet - Special Projects on extranet.atlassian.com	Approved on Jul 15, 2016 at 12:01 AM	READ AND WRITE ACCESS	Revoke Access
 Atlassian JIRA on jira.atlassian.com	Approved on May 30, 2016 at 1:06 AM	READ AND WRITE ACCESS	Revoke Access

Revoking OAuth access tokens

You can revoke an OAuth access token to deny a Jira gadget access to your Jiradata. When you revoke access, the gadget can only access public data on your Jira instance.

1. To view tokens you have issued, go to your **Profile > Tools > View OAuth Access Tokens**
2. Next to the application whose OAuth access you wish to revoke, click **Revoke Access**.
3. You may be prompted to confirm this action. If so, click **OK**.
4. The gadget's access token is revoked and the Jira gadget can only access public Jiradata.

Requesting apps

The [Atlassian Marketplace](#) website offers hundreds of apps that administrators can install to enhance and extend your Jira applications. If the app request feature is enabled for your instance, you can submit requests for Marketplace apps directly to your administrator.

The 'Atlassian Marketplace for Jira' page presents an integrated view of the Marketplace website from within the Jira user interface. The page offers the same features as the Marketplace website, such as app search and category filtering, but tailors the browsing experience to Jira application users.

This in-product view of the Marketplace gives day-to-day users of the Atlassian applications, not just administrators, an easy way to discover the apps that can help them work. When you find an app of interest, you can submit a request with just a few clicks.

Submitting an app request

1. From anywhere in the application, open your profile menu and choose **Atlassian Marketplace**.
2. In the Atlassian Marketplace page, use the search box to find apps or use the category menus to browse or filter by apps by type, popularity, price or other criteria. You can see what your fellow users have requested by choosing the **Most Requested** filter.
3. When you find an app that interests you, click **Request** to generate a request for your administrator.
4. Optionally, type a personal message to your administrators in the text box. This message is visible to administrators in the details view for the app.
5. When ready, click **Submit Request**.
6. Click **Close** to dismiss the 'Success!' message dialog box.

At this point, a notification appears in the interface your administrators use to administer apps. Also your request message will appear in the app details view, visible from the administrator's 'Find New apps' page. From there, your administrator can purchase the app, try it out or dismiss requests.

Updating an app request

After submitting the request, you can update your message at any time. Click the **Update Request** button next to the listing in the 'Atlassian Marketplace' page to modify the message to your administrator.

The administrator is not notified of the update. However, your updated message will appear, as you have modified it in the details view for the app immediately.

Using keyboard shortcuts


Keyboard shortcuts are a great way for you to speed up editing, navigating, and for performing actions without having to take your fingers off the keyboard.

Some keyboard shortcuts require additional permissions or applications, and depend on how your Jira administrator(s) have configured permissions for your user account and which applications are installed.

On this page:

- [View keyboard shortcuts](#)
- [Enabling and disabling keyboard shortcuts](#)

View keyboard shortcuts

- Choose the question mark icon () at top right of the screen, then choose **Keyboard shortcuts**.
- When viewing a page, press **Shift + /**.

The Keyboard Shortcuts dialog is displayed and shows commands for the operating system and browser that you are using. The dialog is divided into sections for the following information:

- **Global shortcuts**- shortcuts that can be used when you are in any part of Jira
- **Navigating issues** - shortcuts for navigating through issues
- **Issue actions** - shortcuts for working with issues
- **App specific**- any application-specific shortcuts. These shortcuts only work in the listed application.

If you have other Jira applications installed, you may have additional keyboard shortcuts available. For example, if you have Jira Software installed, you will see a series of additional keyboard shortcuts in the lower-right of this dialog box (and some additional **Global** keyboard shortcuts specific to Jira Software in the upper-left section). However, the keyboard shortcuts in the **Agile Shortcuts** section only function in Jira Software, and not in a Jira context.

Enabling and disabling keyboard shortcuts

Keyboard shortcuts are enabled by default. However, you can disable them on a per-user basis in the Keyboard Shortcuts dialog box.

1. Ensure you are logged in and open the Keyboard Shortcuts dialog box (see [above](#)).
2. At the bottom of the Keyboard Shortcuts dialog box, click **Disable Keyboard Shortcuts** or **Enable Keyboard Shortcuts**.

You can also disable or re-enable keyboard shortcuts by editing the **Preferences** section of your user profile. See [Managing your user profile](#) for more information.

Modifier keys

Some keyboard shortcuts require modifier keys to be pressed simultaneously, along with a single 'action' key. Modifier keys may differ, depending on your combination of operating system and web browser. The following table identifies the modifier keys for some supported web browsers and operating systems:

Web Browser	Mac OS X	Windows	Linux /Solaris	Notes
Firefox	Ctrl	Alt + Shift	Alt + Shift	In Firefox, it is possible to customize 'Modifier key shortcuts'. Please read Mozilla's documentation for more information.
Internet Explorer		Alt		Typing a 'Modifier key shortcut' that leads to a link requires you to press the 'Enter' to complete the action.
Safari	Ctrl + Alt /Option	Ctrl		

Chrome	Ctrl + Alt /Option	Alt + Shift	Alt + Shift	
--------	--------------------------	-------------	----------------	--

Adding announcements

Do you need to communicate a system outage, post a message about operating hours, or perhaps offer a prompt at login? Announcements are a convenient way to give your customers important information.

On this page:

- [Help center announcements](#)
- [Login screen announcements](#)
- [Customer portal announcements](#)

Here are all the places within Jira Service Management that you can add an announcement:

Help center announcements

The help center is a collection of all your service projects (customer portals). Customers who go to the help center can see all the service projects they have permission to send requests to. To add an announcement here:

1. Go to **Jira administration > Applications > Jira Service Management > Configuration**.
2. Below **Help center**, select the link **View and customize**.
3. On the right hand side, in the customize panel, edit the following fields:
 - **Announcement subject** (for example, Hours of operation)
 - **Announcement message** (for example, Mon-Fri: 09:00 to 17:00)
4. Choose **Save changes** to publish your announcement.

The screenshot shows the 'Add help center announcements' configuration page. It features a left sidebar with a 'Requests' button showing '30' and a user profile icon. The main content area is divided into three sections: 'Add help center announcements', 'Add login announcement', and 'Customize help center'. Each section has input fields for 'Announcement subject', 'Announcement message', and 'Links [name|http://example.com]'. The 'Add help center announcements' section has a blue arrow pointing to its title. The 'Customize help center' section includes fields for 'Help center name' and 'Front page title'. At the bottom, there are 'Save changes' and 'Cancel' buttons.

You can give your agents permission to add announcements to the help centre.

1. Go to **Jira administration > Applications > Jira Service Management > Configuration**.
2. Under the help center setting **Can agents add announcements to the help center?** choose **Yes**.

Login screen announcements

These announcements are displayed to customers who haven't logged in. The announcement might be a useful login prompt, or contact information in cases of difficulty. To add an announcement to the login screen:

1. Go to **Jira administration** > **Applications** > **Jira Service Management** > **Configuration**.
2. Below **Help center**, select the link **View and customize**.
3. On the right hand side, in the customize panel, under **Add login announcement**, edit the following fields:
 - **Announcement subject** (for example, How to login)
 - **Announcement message** (for example, Use your employee login as follows...)
4. Choose **Save changes** to publish your announcement.

The screenshot shows the configuration interface for help center announcements. It features a sidebar on the left with a 'Requests' count of 30 and a user profile icon. The main content area is divided into three sections:

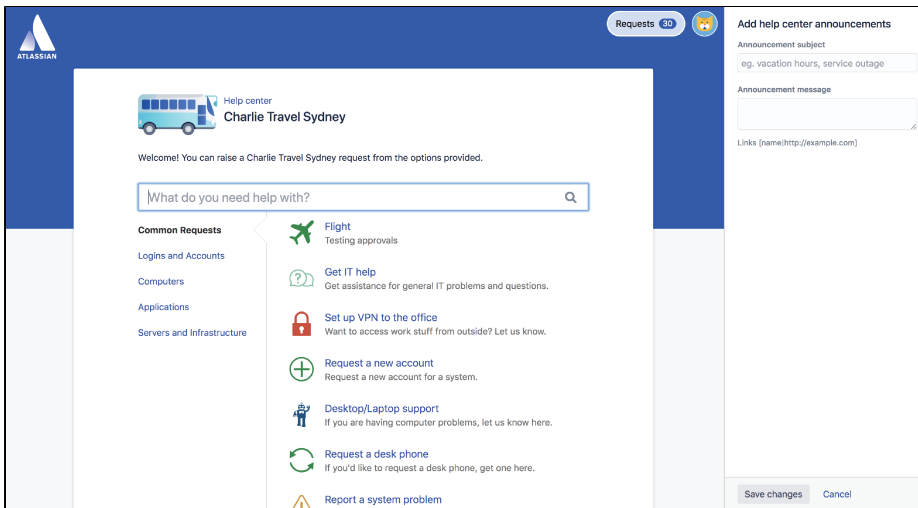
- Add help center announcements:** Includes fields for 'Announcement subject' (Hours of operation), 'Announcement message' (Mon-Fri: 09:00 to 17:00, Sat-Sun: 09:00 to 12:30), and 'Links [name|http://example.com]'.
- Add login announcement:** This section is highlighted with a blue arrow. It includes fields for 'Announcement subject' (e.g. login support, signup instructions), 'Announcement message' (a text area), and 'Links [name|http://example.com]'.
- Customize help center:** Includes fields for 'Help center name' (Help center) and 'Front page title' (Welcome to the help center).

At the bottom of the form are 'Save changes' and 'Cancel' buttons.

Customer portal announcements

Every service project comes with an online portal that your customers use to interact with your service team. To add an announcement to a specific customer portal:

1. Navigate to the service project you want to update.
2. Choose **Project settings** > **Portal settings**.
3. Click the link **View and change** under the **Announcement** heading.
4. On the right hand side, in the customize panel, under **Add help center announcement**, edit the following fields:
 - **Announcement subject** (for example, Holiday hours)
 - **Announcement message** (for example, The IT team will be out of the office from 20th December to 5th January)
5. Choose **Save changes** to publish your announcement.



You can give your agents permission to add announcements to the customer portal.

1. Go to **Project settings > Portal settings**.
2. Under the announcement setting **Can agents add announcements to this portal?** choose **Yes**.

Adding request participants

Request participants are customers and [organizations](#) who can view, comment, and receive notifications about a request. Participants receive the same notifications as the reporter and can turn off notifications at any time. You might add request participants so they can provide more information about a request, or to notify them of a request's progress.

Both agents and customers can add and remove request participants. Who customers can add as participants depends on the project's customer permissions settings. Read [Managing access to your service project](#) to learn more.

On this page:

- [Add participants to a project](#)
- [Add participants to an issue](#)
- [Add participants in the customer portal](#)
 - [Add participants by mentioning them via the customer portal](#)
- [Add participants through email](#)
- [Add watchers](#)

Add participants to a project

To add participants to a project:

1. From the sidebar, select **Project settings > Users and roles**, then select **Add users to a role**.
2. Enter the participant's email address, choose their role type, then select **Add**.

i Request participants follow issue-level security schemes. For example, if an administrator customizes requests so that only reporters can view them, then request participants won't be able to view the request. Administrators can refer to the instructions in [Configuring issue-level security](#) to update the issue security scheme.

Add participants to an issue

Requests participants can only be added if they have access to the project. Change your [customer permissions](#) to set who can access your service desk project.


To add a participant to a request:


1. Open the issue you want to add a request participant to.
2. Select the **Request participants** or **Organizations** field on the right side of the issue view.
3. Enter the participant's details, then click the tick.


When you add a request participant they receive an email notifying them that they are participating in the request, depending on the [project's notification settings](#).


The following image shows the request participants in the issue view:

› **People**

Assignee:  Karen Bywater

Reporter:  Mitch Davis

Request participants:  Alana Grant

Organizations:  Charlie Travel

Votes: 0 [Vote for this issue](#)

Watchers: 1 [Start watching this issue](#)

Add participants in the customer portal

Customers in an [organization](#) can share requests when they raise them. If a customer belongs to only one organization, the request is shared with it by default unless you have changed the default settings to keep all new requests private. If the customer is in more than one organization, the request is private by default.

i You can choose whether new requests are private or shared by default. To do so, go to **Administration > Applications > Jira Service Management configuration**, and look at Organization management.

If you want to share a request with only some participants, you can add them to your request.

How an agent adds a participant to a request:

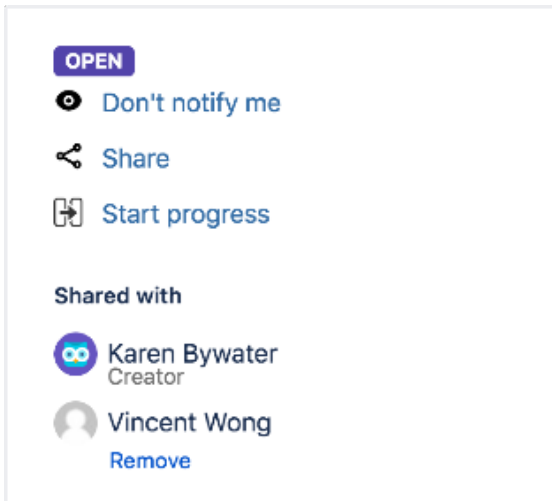
1. From the sidebar, select **Raise a request**.
2. Fill out the request form, select **Create**.
3. Select **Share** from the right side panel, enter the name, email address, or organization in the **Share this request** field.
4. Select **Share**.

How a customer adds a participant to a request:

1. Access the customer portal from the link provided.
2. Fill out the request form, select **Create**.
3. Select **Share** from the right side panel, enter the name, email address, or organization in the **Share this request** field.
4. Select **Share**.

Participants will receive an email notifying them that they are participating in the request depending on the [project's notification settings](#). The participants will appear under the **Shared with** section in the request.


The following image shows the request participants in the customer portal request view:



Add participants by mentioning them via the customer portal


You can also add participants by mentioning them in the description or comments. They will be immediately added as a participants and will receive a notification about it (if the *Participant added* notification is enabled).

- Agents: You can mention anyone in the project.
- Customers: You can only mention customers from your organization. In some projects where customers are allowed to share requests with any customer or organization, you'll be able to mention anyone in the project.

 Actions via customer portal won't result in any Jira mention notification emails.

Add participants through email

Customers in an [organization](#) can share requests when they raise them. If a customer belongs to only one organization, the request is shared with it by default unless you have changed the default settings to keep all new requests private. If the customer is in more than one organization the request is private by default.

 You can choose whether new requests are private or shared by default. To do so, go to **Administration > Applications > Jira Service Management configuration**, and look at Organization management.

If you want to share a request with only some participants, you can add them to your request.

To add participants through email:

1. Add the participant's email address to either the TO or CC fields.

The participant will receive an email notification that you added them depending on the [project's notification settings](#).

Add watchers

Agents can involve other agents or Jira users to get help with an issue. For example, you might want Jira Software developers to help analyze a bug that a customer has reported.

To involve other users, add them as watchers:

1. Open the issue you want to add a watcher to.
2. Select the **Watchers** field to the right side, under **People**.
3. Select the number count, then enter the name or email address in the **Add watchers** field.

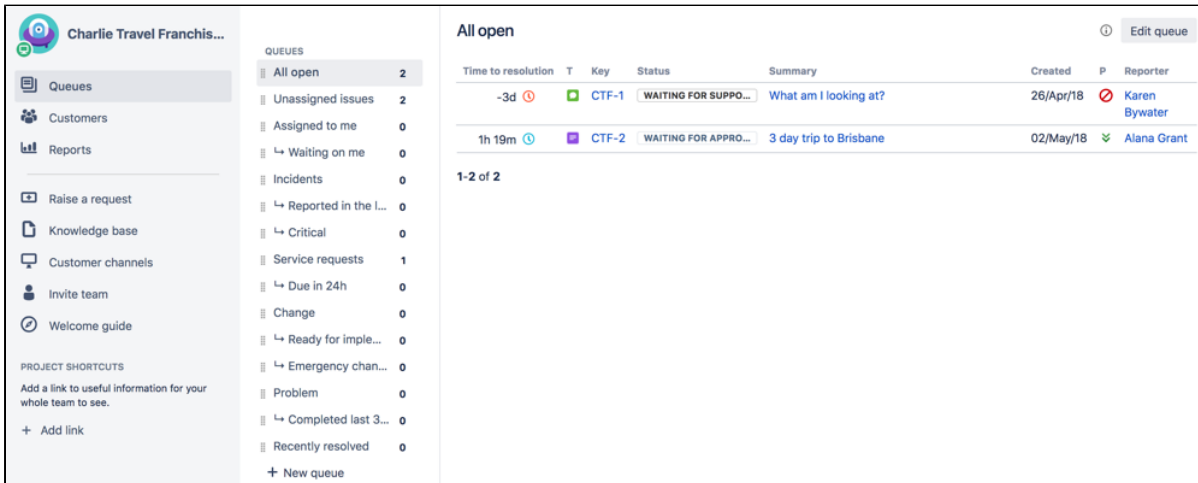
The screenshot shows the 'People' section of a Jira issue. It includes fields for Assignee (Unassigned), Reporter (Alana Grant), Request participants (None), Approvers (Karen Bvwater), Organizations (None), Votes (0), and Watchers (1). An 'Add Watchers' dialog box is open, showing a search input field and a list of users, with 'Karen Bywater' selected. Below the Watchers field, there is a 'WAITING FOR APPRO...' badge and a '1 approval needed' indicator with a user profile for Karen Bywater.

Watchers are notified about internal activity on an issue, and can communicate with you via internal comments.

Using service project queues



Customer requests become issues that you can view and work on in queues. Jira Service Management comes with default queues that your administrator can update to automatically triage issues for your team. As an agent, you can see how many issues are in each queue, and switch between queues to work on the right issues at the right time.

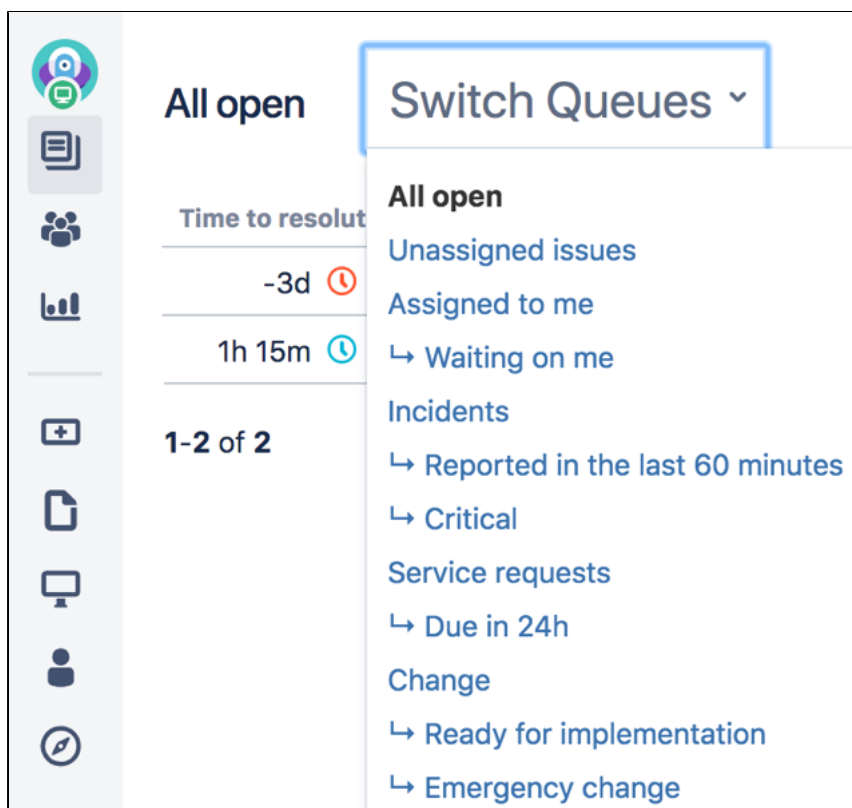
You can easily navigate to your service project queues at any time by selecting **Queues** from your project sidebar.



Switching queues

When you select **Queues** from your project sidebar for the first time, the secondary sidebar menu will open automatically. This sidebar displays all queues in your service project, as well as the number of issues in each queue. Simply select the name of the queue you wish to work from to view its issues.

To expand the view of a single queue, you can minimize your project sidebar by selecting **Collapse sidebar** () and minimize your queue sidebar by selecting  in the sidebar's upper right corner. When the queue sidebar is collapsed, a Switch queue dropdown will appear, which you can use to view a different queue or to reopen the queue sidebar:



Sorting issues

To make it easier to find the most important issues, you can sort issues in your queues based on any of the available columns. Click a column to choose it as the primary sorting option, and then click again to refine the sorting. For example, you can change the sorting from oldest to newest, or switch the alphabetical order.

All open								
Sorted shortest → longest (Default)								
Time to resolution ↑	T	Key	Status	Summary	Created	Reporter	Due	
3h 36m		CHARLIE-1	WAITING FOR SUPPORT	What am I looking at?	30/Oct/19	John Williams		
1d 7h		CHARLIE-2	ESCALATED	Low quality rocket fuel	30/Oct/19	Mr. Jack		
1d 12h		CHARLIE-3	PENDING	Engine makes weird noises	30/Oct/19			
2d 1h		CHARLIE-5	IN PROGRESS	Cleaning service in the Mars base	30/Oct/19	Jimmy Hawking		
2d 7h		CHARLIE-6	WAITING FOR SUPPORT	Red dust gets everywhere	30/Oct/19	Suzy Queue		
2d 12h		CHARLIE-7	WAITING FOR SUPPORT	Mayday, I'm lost in space	30/Oct/19	Ann		
3d 2h		CHARLIE-8	WAITING FOR SUPPORT	When can we expect travels to Saturn?	30/Oct/19	Walter Gray		
3d 6h		CHARLIE-9	WAITING FOR SUPPORT	Requesting discount for Moon tickets	30/Oct/19	Jason Wild		
3d 7h		CHARLIE-4	IN PROGRESS	Requesting more flights to Mars	30/Oct/19	Alfred Pear		

Some things to keep in mind:

- Sorting is a personal setting. Whatever you choose, it won't affect your teammates.
- Your settings are saved in the local storage, so they won't be available if you access your queues from a different device.
- To reset sorting, select **More** (⋮) > **Reset sorting**.

Managing multiple issues at once

You can update multiple issues at once instead of opening and editing them one by one.

The screenshot shows a Jira issue list in 'All open' view. At the top, there are 6 selected issues. A toolbar contains actions: Link, Comment, Assign, Watch, and Delete. The issues are listed in a table with columns for selection, issue key, status, description, created date, and due date.

	Issue Key	Status	Description	Created	Due
<input type="checkbox"/>	CHARLIE-1	WAITING FOR SUPPORT	What am I looking at?	30/Oct/19	
<input checked="" type="checkbox"/>	CHARLIE-4	IN PROGRESS	Requesting more flights to Mars	30/Oct/19	
<input checked="" type="checkbox"/>	CHARLIE-2	ESCALATED	Low quality rocket fuel	30/Oct/19	
<input checked="" type="checkbox"/>	CHARLIE-3	PENDING	Engine makes weird noises	30/Oct/19	
<input type="checkbox"/>	CHARLIE-9	WAITING FOR SUPPORT	Requesting discount for Moon tickets	30/Oct/19	
<input type="checkbox"/>	CHARLIE-7	WAITING FOR SUPPORT	Mayday, I'm lost in space	30/Oct/19	
<input checked="" type="checkbox"/>	CHARLIE-8	WAITING FOR SUPPORT	When can we expect travels to Saturn?	30/Oct/19	
<input checked="" type="checkbox"/>	CHARLIE-5	IN PROGRESS	Cleaning service in the Mars base	30/Oct/19	
<input checked="" type="checkbox"/>	CHARLIE-6	WAITING FOR SUPPORT	Red dust gets everywhere	30/Oct/19	

1 - 12 of 12

After selecting multiple issues, you'll be able to choose one of these actions:

- **Assign** issues to a different user
- **Watch / Stop watching** selected issues. This requires that watching is enabled in your Jira instance.
- **Delete** selected issues
- **Add an internal or external comment** to your issues. Adding a comment to 50 issues at once might take some time, so watch out for this.
- **Link** selected issues to other issues

Good to know

- You can select a maximum of 50 issues at once because that's what fits on a single page. When you switch to another page, your selections will be cleared.
- Keyboard shortcuts: You can use *up* and *down* arrows to move between issues, and the *space bar* to select them.


Raising requests on behalf of customers

Let's say you're helping a customer resolve an issue over the phone, and you need to follow up with additional information. You can use the customer portal to quickly enter your customer's name, fill in the issue details, and submit the service project request.


If you don't need to create a request but simply want to invite customers to your service project so they know how to get help, you can skip ahead to [Invite a new customer](#).

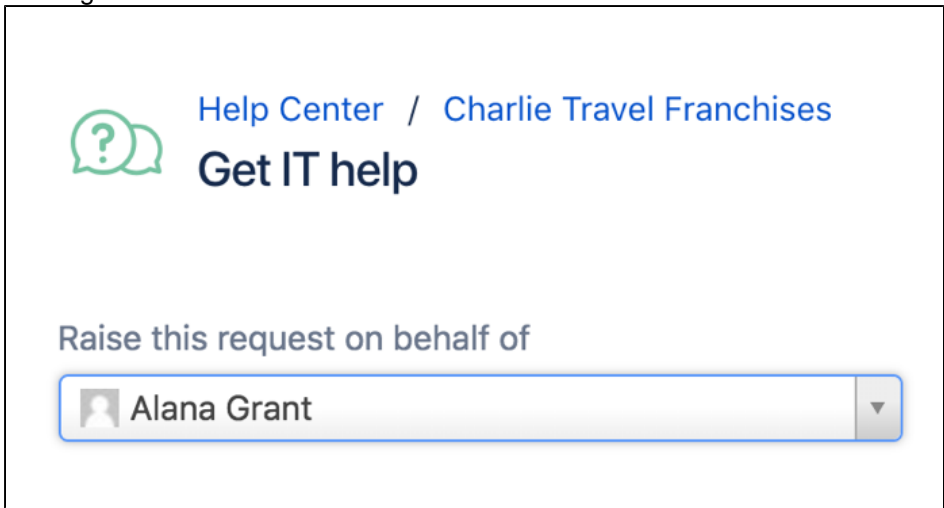
On this page:

- [Raise a customer request](#)
- [Invite a new customer](#)

 Only agents can raise requests on behalf of customers.

Raise a customer request

1. From your service project sidebar, select **Raise a request** ().
2. Select the request type that matches your customer's need.
3. In the **Raise this request on behalf of** field, enter a new customer's email address, or search for an existing customer:



4. Fill in the request details. If the customer is in an organization, you can choose to share the request with that organization.
5. Select **Create**. Your customer is emailed a link to the new request. New customers also receive an invitation to finish creating a service desk account.

Invite a new customer

1. From your service project sidebar, select **Customers**.
2. Select **Add customers** and enter your customer's email address.
3. If they belong to an organization, select it from the **Add to organization** dropdown.
4. Select **Add**, and you're done!

Adding customers


As an agent, you can add customers to your service project. If your administrator grants you permission, you can also manage organizations in the project.

- **Customers** raise requests in your service desk.
- **Organizations** are groups of customers that are shared across projects.

On this page:


- [Add customers](#)
- [Add organizations](#)
- [Add customers to an organization](#)
- [Remove customers from an organization](#)
- [Remove organizations from a project](#)

Add customers


Add customers to a project via **Customers**  **Add customers**. Customers on this list can raise requests in the project, via the portal or email.

Customers are automatically added to the list if your service project is open to users with Jira accounts, or allows customers to create their own accounts.

Add organizations

Add a new or existing organization to a project via **Customers**  **Add organizations**. Organizations that you add display on the **Customers** list, and their members can raise requests in the project via the portal or email.

Add customers to an organization

1. Go to the **Customers** .
2. Select the organization in the list.
3. Click **Add customers**. The customers you add can raise requests in all projects that use the organization.

Because the organization can be used in multiple projects, the customers are not added to the **Customers** role for the project.

Remove customers from an organization

1. Select an organization in the **Customers** list.
2. Find the customer you want to remove, and then click **X** next to their name.

Customers you remove lose access to projects that use the organization, unless they have access through another organization, or have the **Customers** role for the projects.

Remove organizations from a project

1. Select an organization in the **Customers** list.
2. Select **Remove from project**.

When you remove an organization from a project, its members lose access to the project unless they have the **Customers** role for the project.

Organizing work with versions

Versions are points-in-time for a project. They help you organize your work by giving you milestones to aim for. You can then assign the issues in your project to a specific version, and build up the work you need to do to complete that version.

On this page:

- [Managing a project's versions](#)
- [Add a new version](#)
- [Release a version](#)
- [Archive a version](#)
- [Delete a version](#)
- [Merge multiple versions](#)
- [Reschedule a version](#)

You need to have the project-specific **Administer Projects** [project permission](#) or the **Jira Administrator** [global permission](#) to be able to:

- Add create a new version against which issues can be aligned.
- Release mark a version as released.
- Archive hide an old version from the Releases report, and in the user interface.
- Delete remove a version. You must choose an action for any issues with that version.
- Merge combine multiple versions into one.
- Reschedule re-arrange the order of versions.

Once a version has been created for a project, the 'Affects version' and 'Fix version' fields will become available for your issues. If you cannot see these fields on your issue, your project may not have any version yet, or the fields are hidden from view.

Managing a project's versions

The easiest way to manage a project's versions is through the Versions page.

1. Choose **Administration** (⚙️) > **Projects**, and click the name of the project.
2. Choose **Versions** in the sidebar. The **Versions** page is displayed, showing a list of versions.

Screenshot: The 'Versions' page

Name	Description	Start date	Release date
2.1	Bug fix and feature polish		27/Mar/13
3.0	5 New Levels, Android Support		17/Feb/13
2.0	UI cleanup		27/Mar/13
1.3	The Phantom Nerd		29/Jul/11
1.2	Return of the Nerds		15/Jun/11
1.1	The Empire Strikes Nerds		08/Jun/11
1.0	A New Hope for Nerds		01/Apr/11


Add a new version

1. The Add Version form is located at the top of the 'Versions' page.
2. Enter the name for the version. The name can be:
 - simple numeric, e.g. "2.1", or

- complicated numeric, e.g. "2.1.3", or
 - a word, such as the project's internal code-name, e.g. "Memphis".
3. Optional details such as the version description (text not HTML), start date and release date (i.e. the planned release date for a version) can also be specified. These can be changed later if required.
 4. Click the **Add** button. You can drag the new version to a different position by hovering over the 'drag' icon at the left of the version name.

Release a version

1. On the 'Versions' page, hover over the relevant version to display the cog icon, then select **Release** from the drop-down menu.
2. If there are any issues set with this version as their 'Fix For' version, Jira allows you to choose to change the 'Fix For' version if you wish. Otherwise, the operation will complete without modifying these issues.

 To revert the release of a version, simply select **Unrelease** from the drop-down menu.

Archive a version

1. On the 'Versions' page, hover over the relevant version to display the cog icon, then select **Archive** from the drop-down menu.
2. The version list indicates the version 'archived' status with a semi-transparent icon. No further changes can be made to this version unless it is un-archived. Also it is not possible to remove any existing archived versions from an issue's affected and fix version fields or add any new archived versions.

 To revert the archive of a version, simply select **Unarchive** from the drop-down menu.

Delete a version

1. On the 'Versions' page, hover over the relevant version to display the cog icon, then select **Delete** from the drop-down menu.
2. This will bring you to the 'Delete Version: <Version>' confirmation page. From here, you can specify the actions to be taken for issues associated with the version to be deleted. You can either associate these issues with another version, or simply remove references to the version to be deleted.


Merge multiple versions

Merging multiple versions allows you to move the issues from one or more versions to another version.

1. On the 'Versions' page, click the **Merge** link at the top right of the page.
2. The 'Merge Versions' popup will be displayed. On this page are two select lists both listing all un-archived versions.
In the 'Merging From Versions' select list, choose the version(s) whose issues you wish to move. Versions selected on this list will be removed from the system. All issues associated with these versions will be updated to reflect the new version selected in the 'Merge To Version' select list. It is only possible to select one version to merge to.
3. Click the **Merge** button. If you are shown a confirmation page, click **Merge** again to complete the operation.

Reschedule a version

Rescheduling a version changes its place in the order of versions.

- On the 'Versions' page, click the  icon for the relevant version, and drag it to its new position in the version order.

Workflows

All Jira projects contain issues that your team can view, work on, and transition through stages of work from creation to completion. The path that your issues take is called a workflow. Each Jira workflow is composed of a set of statuses (the state your work can be in) and transitions (how your work moves between statuses) that your issue moves through during its lifecycle, and typically represents work processes within your organization.

In addition, Jira uses workflow schemes to define the relationship between issue types and workflows. Workflow schemes are associated with a project, and make it possible to use a different workflow for different combinations of project and issue types.

Jira administrators and project administrators have different permissions when it comes to workflows.

Project administrators

As a project administrator, you can only edit a workflow that belongs to your project if:

- you have the *Extended project administration* permission, which is enabled by default (you can check that in **Project settings > Permissions**),
- the workflow isn't shared with any other projects (it's only available in your project),
- the workflow isn't the default Jira workflow (no-one can edit these workflows).

If the workflow is shared with another project, you'll see that information when you view the workflow. You'll also see how many issue types share the workflow, and would be affected by any changes you may make. You can make the following changes to the workflow:

- Add a status (the statuses must already exist in the Jira instance, you can't create, edit or remove statuses),
- Delete a status (the statuses must not be used by any of the project's issues),
- Create, add, edit or delete transitions (you can't select or update a screen used by the transition, or edit or view a transition's properties, conditions, validators or post-functions).

To view a workflow

1. Select **Projects** and choose the project whose workflow/s you want to view.
2. Select **Project settings** in the sidebar.
3. Select **Workflows** to see the list of workflows and issue types they're associated with.
 - Click a workflow to display it as diagram. If you're able to edit the workflow, you'll see an **Edit** button. If the workflow is shared with another project or issue type/s, that information will be available, and you can view it by clicking the relevant link.
 - Additionally, you can view a workflow in a simple, text form by clicking **View as text** next to the workflow's name.

To edit a workflow

1. When viewing a workflow, select **Edit**.
2. You can add a status or transition by clicking the relevant button. You can edit existing transitions by selecting them.
3. **Publish** your workflow to make it active.

If you don't publish the workflow, it'll remain as a draft until such time as you publish it, or discard it.

If you have a draft workflow present on your project, and you want to see the original workflow that's currently active, select **Project settings > Workflows**, and click a workflow.

Jira administrators

As a Jira administrator, you can complete the actions listed in the table below. The actions you have available are more extensive, and the documentation links will direct you to the Administrator documentation set.

What you can do...	Documentation
--------------------	---------------

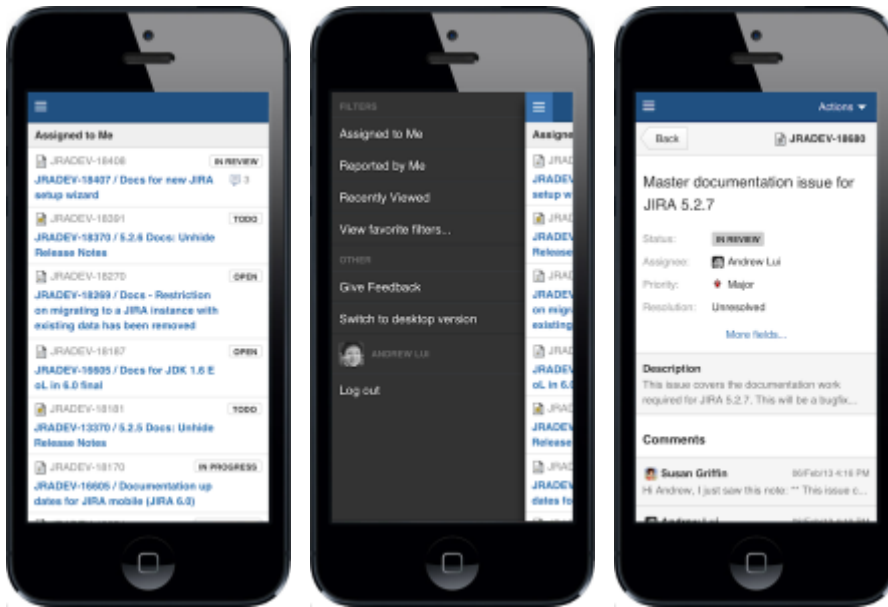
<ul style="list-style-type: none">• Edit existing workflows• Create new workflows• Configure existing workflows	Working with workflows
<ul style="list-style-type: none">• Add a workflow scheme• Configure a workflow scheme• Manage workflow schemes	Configuring workflow schemes
<ul style="list-style-type: none">• Import and export workflows• Activate and deactivate workflows	Managing your workflows
<ul style="list-style-type: none">• Add custom events• Configure the initial status• Work in text mode• Configure workflow triggers• Use validators and custom fields• Use XML to create a workflow• Workflow properties	Advanced workflow configuration

Using Jira on a mobile device

When you view a Jira page on a mobile device, such as an iPhone or an Android phone, Jira will display an optimized version of the page. Jira chooses the mobile or desktop interface based on your device.

The Jira mobile interface is designed for viewing and interacting with issues on the go. If you need full access to Jira, you can always switch to the Jira desktop interface via the mobile menu (shown in the screenshots below).

What does Jira look like on a mobile device?



What can you do in Jira on a mobile device?

The Jira mobile interface has been designed to give users quick access to their issues on the go. This includes;

- Viewing issues, comments, attachments, issue links and your favorite filters.
- Performing basic operations like adding comments, watching or voting on issues and assigning issues to users.

If you need to create or modify issues on the go, you can still do so by switching to the desktop interface via the mobile menu (shown in the screenshots above).

Configuring dashboards

Your dashboard is the main display you see when you log in to your project. You can create multiple dashboards for different projects, or multiple dashboards for one big project. Each project has a default dashboard, or you can create a personal dashboard and add gadgets to keep track of assignments and issues you're working on. Dashboards are designed to display gadgets that help you organize your projects, assignments, and achievements in different charts.

You can see all dashboards by selecting the **Dashboards** drop-down from your Jira application header.

On this page:

- [About the default dashboard](#)
- [Creating a dashboard](#)
- [Managing dashboards and permissions](#)
- [Sharing and editing your dashboard](#)
- [Adding favorite dashboards](#)
- [Note on dashboard permissions](#)
- [Setting up a Wallboard](#)


About the default dashboard

The gadgets on the default dashboard can be reordered and switched between the left and right columns. Additional gadgets can also be added, while some gadgets can be configured. The layout of the dashboard (e.g. number of columns) can also be configured.

All changes made to the default dashboard will also change the dashboards of all users currently using the default dashboard. However, gadgets that users do not have permissions to see will not be displayed to them. For example, the 'Administration' gadget, although it may exist in the default dashboard configuration, will not be visible to non-admin users.

Creating a dashboard

You can easily create and customize your own dashboard to display the information you need. Note that only administrators can customize the default dashboard for your project.

1. At the top right of the Dashboard, select **More** ()
2. Select either **Create Dashboard** to create a blank dashboard, or **Copy Dashboard** to create a copy of the dashboard you are currently viewing.
3. Name and describe your dashboard.
4. Fill out the rest of the fields as applicable.
5. Click **Add**.

By default, sharing is set to private if you have not specified a personal preference. You can adjust this setting in the sharing preferences in your [user profile](#), and change dashboard permissions at any time in the [Manage Dashboards](#) page.

Choosing a dashboard layout

To choose a different layout for your dashboard page (for example, three columns instead of two):

1. At the top right of the Dashboard, click **Edit layout**. A selection of layouts will be displayed:



2. Choose your preferred layout.

Managing gadgets

To get the most out of your dashboard, including adding, rearranging, removing, and configuring gadgets, see [Adding and customizing gadgets](#).

Managing dashboards and permissions

You can edit, delete, copy, mark favorites, and share your dashboards from the Manage Dashboards page.

1. Select **Dashboards > Manage Dashboards**.
2. Choose the dashboard.

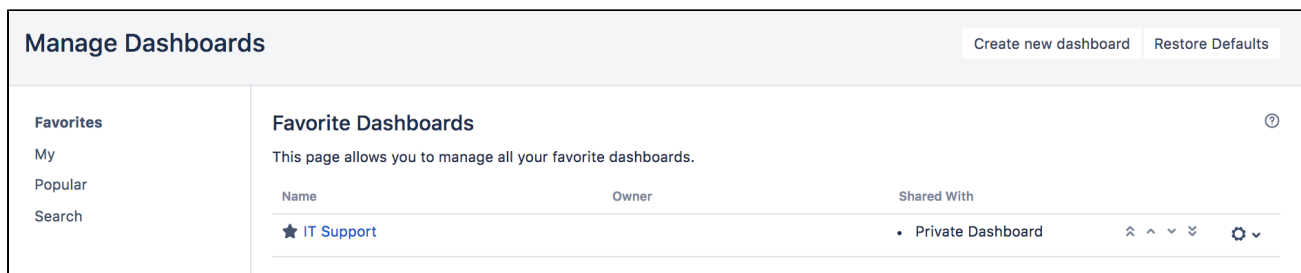
Sharing and editing your dashboard

You can edit the details for your dashboard, and restrict or share with other users according to the permissions that are set. In addition, you can see all the dashboards you've created, any public dashboards, and any shared dashboards.

1. Click **Actions** (⚙️) > **Edit**. (If you're viewing the dashboard, go to **More** (⋮) > **Edit/Share Dashboard**).
2. Edit the settings.

Adding favorite dashboards

If you find a dashboard you like, click the star icon next to its name to add it to your favorite dashboards list. You can also add the default dashboard to your favorites list so it's easily available to you.



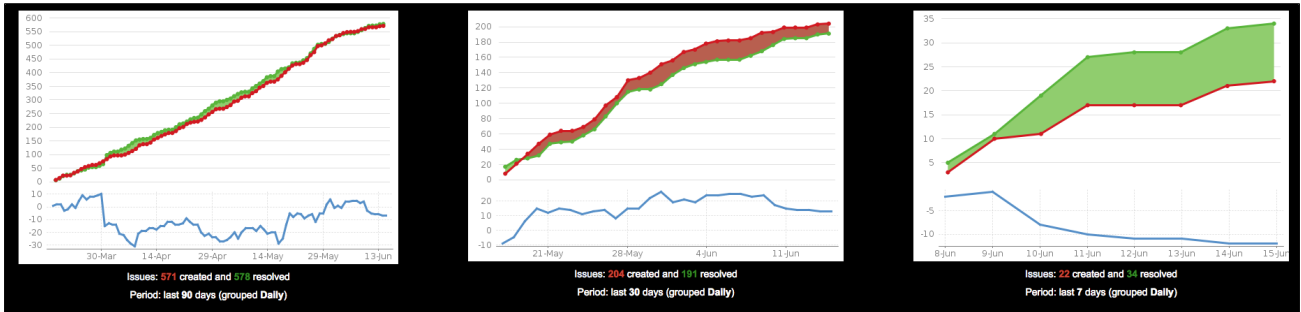
Note on dashboard permissions

Jira administrators, as set in global permissions, can manage their users' shared dashboards in the **Shared dashboards** menu. Administrators can also change the ownership of a dashboard if the creator is unable to maintain the dashboard or its gadgets. See [Managing shared dashboards](#) for more information.

Setting up a Wallboard

Turn any Jira application dashboard into a wallboard by plugging your computer into a TV monitor. The Wallboard is a dashboard gadget that acts as an information radiator to provide instant visual insight into project progress and team accomplishments. With your favorite dashboard selected, select **More** (⋮) > **View as Wallboard**. The dashboard will appear against a black background, and will rotate gadgets if the user enables the slideshow option.

The Wallboard below shows the same **Created vs. Resolved Issues** gadgets and data above.



Adding and customizing gadgets

Adding a gadget to a dashboard

You can add gadgets to your own personal dashboard(s). To add a gadget to the default dashboard for your Jira application, you must be a Jira admin.





Some applications allow dashboards that are shared by groups of people. If you have permission to update a shared dashboard, the other people sharing the dashboard will see your changes, too.

1. Go to the dashboard by selecting the **Dashboards** link in the header.
2. If you don't already have a dashboard, select **Manage Dashboards** from the dropdown, then **Create new dashboard**.
3. Once your dashboard is created, on the dashboard, select **Add Gadget**.
4. Use the gadget wizard to choose the gadgets you want to add. You can see a list of these gadgets in [Gadgets for Jira applications](#).

For more information about managing dashboards, see [Configuring dashboards](#).

Customizing how gadgets look

There are a few ways you can customize the view of gadgets in a dashboard:

To	Do this
Expand or collapse gadgets	Use the More () button in the gadget header.
Expand a gadget to take up the entire dashboard	Use the Maximize () button in the gadget header. This view often provides more functionality than is available in the standard view of the gadget. Only some gadgets provide the maximized or canvas view. The canvas view setting is stored in a cookie, and is not saved to the dashboard server.
Rearrange gadgets	Use the  button in the gadget header.
Customize the gadget frames	Use the More () button in the gadget header.
Delete a gadget	

Custom gadgets

You need administrator privileges to add a gadget to the list of available gadgets. If you have permission to add and remove gadgets from the directory itself, you will see the '**Add Gadget to Directory**' and '**Remove**' buttons on the 'Add Gadget' screen. This functionality is only available for the Server version of applications; if you would like to add an Atlassian gadget to a directory in your Cloud site, please contact Atlassian Support.

Gadgets for Jira applications

Gadgets let you customize the information you display on dashboards or wallboards in Jira applications. This information may include the number of open and resolved issues, labels used in selected projects, version reports, and many more.

Both Jira admins and users can manage gadgets on dashboards. The Jira admin can make changes to the system dashboard, while users can only create and customize their personal dashboards. What the user can see in gadgets on the dashboard depends on [global and project permissions](#).

On this page, you'll find all the gadgets that you can install on dashboards in Jira Core (JC), Jira Software (JSW), and Jira Service Management (JSM), including additional Bamboo, Crucible, and FishEye gadgets.

Gadget	JC	JSW	JSM	Description
Jira Core gadgets				
Assigned to Me	✓	✓	✓	Shows all unresolved issues assigned to you.
Bubble Chart	✓	✓	✓	<p>Tracks the correlation of issues in a project or filter during a configured period based on:</p> <ul style="list-style-type: none"> the number of days issues have been open the number of comments in issues the number of participants or votes in issues <p>The horizontal axis represents the number of days when issues remained open. The vertical axis represents the number of comments in issues.</p> <p>The bubble colors also indicate the correlation between the days when issues remained open and the number of comments. The green indicates low values and the red color indicates high values.</p> <p>Only the first 200 open issues are displayed on the bubble chart.</p> <p>You can configure the following settings for the chart:</p> <ul style="list-style-type: none"> the period during which issue comments are considered recent the basis of the bubble size (participants or votes) automatic refresh of the bubble chart data every 15 minutes relative coloring to distinguish issues with more comments from issues with fewer comments logarithmic scale (the linear scale is the default) to distribute the bubbles from each other. We recommend that you use the logarithmic scale if the bubble chart contains a large range of data.
Created vs. Resolved Chart	✓	✓	✓	<p>Shows the work progress by showing the number of created and resolved issues for a set period.</p> <p>The chart is based on your choice of project or issue filter. The chart can be either cumulative or show the issue count.</p> <p>An issue is marked as resolved in the time period if its resolution date fits into this period.</p> <p>The resolution date is the last date when the Resolution field was set to any non-empty value.</p>
Favorite Filters	✓	✓	✓	Shows a list of all issue filters that you've added as favorite filters.

Filter Results	✓	✓	✓	Shows the results of a selected issue filter on the dashboard.
Introduction	✓	✓	✓	Lets you configure a message on the dashboard to say hello to users. The text or HTML displayed in the gadget should be configured by the Jira admin. To do this: 1. Go to Administration > System . 2. In the left panel, go to General configuration . 3. Select Edit settings . 4. In the Introduction field, type the text that will be displayed as the introduction message on the dashboard. 5. At the bottom of the page, select Update to save the changes.
Issue Statistics	✓	✓	✓	Shows issue statistics returned from a selected project or saved filter (grouped by a selected field).
Issues In Progress	✓	✓	✓	Shows all issues that are currently in progress and assigned to you.
Labels Gadget	✓	✓	✓	Shows a list of all labels used in a selected project.
Pie Chart	✓	✓	✓	Shows aggregated data on issues based on a selected filter.
Quick Links	✓	✓	✓	Adds links to frequently used searches and operations.
Two Dimensional Filter Statistics	✓	✓	✓	Shows data based on a selected issue filter. For example, you can create a filter to retrieve all open issues in a particular project. Then, you can configure the gadget to display the statistical data on this collection of issues in a table with configurable axes.
Voted Issues	✓	✓	✓	Shows all issues you've voted for.
Watched Issues	✓	✓	✓	Shows all issues you're watching.
Atlassian gadgets				
Activity Stream	✓	✓	✓	Shows activity on your instance. (Think of it as of a news feed.)
Average Age Chart	✓	✓	✓	Shows the average age of unresolved issues. This chart is based on your choice of project or issue filter and units of time (hours, days, weeks, months, quarters, or years). For the purposes of this gadget, an issue is defined as unresolved if it has no value in the Resolution field. The age of the issue is the difference between the current date and the date when the issue was created.
Heat Map	✓	✓	✓	Shows the relative weighting of a selected fields values in issues of a selected project or filter. For example, this gadget can be configured to show a heat map of issue priority in a particular project.

Jira Road Map	✓	✓	✓	Shows which versions are due for release in a set period, as well as a summary of the progress made towards completing the issues in the versions.
Projects	✓	✓	✓	Shows information on various issue filters from a selected project.
Recently Created Chart	✓	✓	✓	Show a rate at which issues are being created as well as how many of these issues are resolved.
Resolution Time	✓	✓	✓	Shows trends in the average time taken to resolve issues. The report is based on your choice of project or issue filter and units of time (hours, days, weeks, months, quarters, or years). The resolution time is the difference between the dates when an issue was created and resolved. If the resolution date isnt set, the gadget wont consider this issue. The resolution date is the last date when the Resolution field was set to any non-empty value.
Time Since Chart	✓	✓	✓	Show the number of issues for which a selected field was set on a selected date (for example, Created , Updated , Due , Resolved , or a custom field). The Resolved field here is the system Resolution Date field. This is the last date when the Resolution field was set to any non-empty value. The report is based on your choice of project or issue filter and units of time (hours, days, weeks, months, quarters, or years).
Wallboard Spacer Gadget	✓	✓	✓	Lets you modify the layout of your wallboards by adding custom space between gadgets. You should set the height of a spacer in pixels and place the gadget in the preferred position on the dashboard. You cant modify the gadgets color as it matches the black background color in the wallboard view.
Jira Software gadgets				
Agile Wallboard Gadget		✓		Helps you see how youre tracking with an agile board on your wallboard or dashboard.
Days Remaining in Sprint Gadget		✓		Shows how many working days you have before the current sprint ends.
Sprint Burndown Gadget		✓		Shows the burndown for a sprint in a line chart. The vertical axis represents the estimation statistic . The gadget will only display uncompleted sprints .

Sprint Health Gadget	✓	✓	✓	<p>Shows a summary of issues in a sprint in a color-coded bar graph. The colors in this gadget match the colors in the column configuration.</p> <p>The completed work is calculated based on the estimation statistic for a board and is represented with the green part of the progress bar. For example, if you have 50 story points in a sprint and you have 3 resolved issues with 10 story points, the Work complete will be 20% (10 out of 50 story points).</p> <p>The gadget doesn't show the progress of work logged in the Remaining Estimate and Time Spent fields in Jira.</p> <p>Adding or removing an issue from a started sprint is considered a change of scope. The percentage is calculated by using the statistic configured for the board. For example, if you started a sprint with 50 story points and add an issue with 5 story points, the Sprint Health gadget will show a 10% scope change.</p> <p>If you add or remove issues without estimates, the scope change won't be altered.</p> <p>If you're using Jira's time tracking, the scope change won't show.</p> <p>The Blocker field counts all blockers with the statuses To Do or In Progress.</p> <p>For more information, see Configuring estimation and tracking</p>
Version Report	✓	✓	✓	Tracks a projected release date for a version. This helps you monitor whether the version will release on time so you can take action if the work is falling behind.
Jira charting gadgets				
Average Number of Times in Status	✓	✓	✓	Shows the average number of times when issues had a selected status for a set period. Requires the Jira Charting App installed.
Average Time in Status	✓	✓	✓	Shows the average time that issues have spent in a selected status for a set period. Requires the Jira Charting App installed. Learn more about how to use the gadget
Time to First Response	✓	✓	✓	Displays the number of hours that it takes to respond to issues in a project or filter. Requires the Jira Charting App installed.
Workload Pie Chart	✓	✓	✓	Uses a pie chart to show matching issues for a project or filter. Requires the Jira Charting App installed.
Other gadgets				
Jira Issues Calendar	✓	✓	✓	Generates a calendar-based view of due dates for issues and versions. Requires the Jira Calendar app installed.
Test Sessions	✓	✓	✓	Shows a list of test sessions.

Text	✓	✓	✓	<p>Shows HTML text on the dashboard.</p> <p>This gadget is available if a Jira admin has enabled it. By default, the gadget is disabled because it causes a potential security risk. The gadget may contain arbitrary HTML which potentially may make your Jira system vulnerable to XSS attacks.</p> <p>To enable the gadget:</p> <ol style="list-style-type: none"> 1. Go to Administration > Manage apps. 2. In the left panel, select Manage apps to view the apps installed on the Jira site. 3. In the dropdown list, select System. 4. Find and select Atlassian Jira - Plugins - Gadgets Plugin. 5. Find the Text module and select Enable. <p>If you can't enable the gadget, contact Atlassian Support for assistance.</p>
Bamboo gadgets				
Bamboo Charts	✓	✓	✓	<p>Shows Bamboo plan statistics in your dashboard.</p> <p>If you want to add the gadget to the dashboard, a Jira admin should configure the Bamboo plugin on a Jira server. If you have multiple Bamboo servers added in Jira, only one Bamboo Charts gadget will be available per server.</p> <p>When you add the gadget to the Jira dashboard, you may see the following message:</p> <p>The website (container) you have placed this gadget on is unauthorized. Please contact your system administrator to have it approved.</p> <p>To fix this, you should configure the Bamboo site to allow Jira to draw information from it via gadgets on the dashboard.</p> <p>To do this, the Jira admin should first define the Jira site as an OAuth consumer in Bamboo. Then, you should perform a once-off authentication so that the gadget displays correctly.</p>
Bamboo Plan Summary Chart	✓	✓	✓	<p>Shows a graphical summary of a Bamboo build plan.</p> <p>If you want to add the gadget to the dashboard, a Jira admin should configure the Bamboo plugin on the Jira server. If you have multiple Bamboo servers added in Jira, only one Bamboo Plan Summary Chart gadget will be available per server.</p> <p>When you add the gadget to the Jira dashboard, you may see the following message:</p> <p>The website (container) you have placed this gadget on is unauthorized. Please contact your system administrator to have it approved.</p> <p>To fix this, you should configure the Bamboo site to allow Jira to draw information from it via gadgets on the dashboard.</p> <p>To do this, the Jira admin should first define the Jira site as an OAuth consumer in Bamboo. Then, you should perform a once-off authentication so that the gadget displays correctly.</p>

<p>Bamboo Plans</p>	<p>✓</p>	<p>✓</p>	<p>✓</p>	<p>Shows a list of all plans on a particular Bamboo server and the current status of each plan.</p> <p>If you want to add the gadget to the dashboard, a Jira admin should configure the Bamboo plugin on the Jira server. If you have multiple Bamboo servers added in Jira, only one Bamboo Plans gadget will be available per server.</p> <p>When you add the gadget to the Jira dashboard, you may see the following message:</p> <p>The website (container) you have placed this gadget on is unauthorized. Please contact your system administrator to have it approved.</p> <p>To fix this, you should configure the Bamboo site to allow Jira to draw information from it via gadgets on the dashboard.</p> <p>To do this, the Jira admin should first define the Jira site as an OAuth consumer in Bamboo. Then, you should perform a once-off authentication so that the gadget displays correctly.</p>
<p>Clover Coverage</p>	<p>✓</p>	<p>✓</p>	<p>✓</p>	<p>Shows the clover coverage of plans from a particular Bamboo server.</p> <p>If you want to add the gadget to the dashboard, a Jira admin should configure the Bamboo plugin on the Jira server. If you have multiple Bamboo servers added in Jira, only one Clover Coverage gadget will be available per server.</p> <p>When you add the gadget to the Jira dashboard, you may see the following message:</p> <p>The website (container) you have placed this gadget on is unauthorized. Please contact your system administrator to have it approved.</p> <p>To fix this, you should configure the Bamboo site to allow Jira to draw information from it via gadgets on the dashboard.</p> <p>To do this, the Jira admin should first define the Jira site as an OAuth consumer in Bamboo. Then, you should perform a once-off authentication so that the gadget displays correctly.</p>
<p>FishEye gadgets</p>				
<p>Crucible Charts</p>	<p>✓</p>	<p>✓</p>	<p>✓</p>	<p>Shows statistical summaries of your code reviews.</p> <p>If you want to add the Crucible Charts gadget to your dashboard, a Jira admin should configure the FishEye app on your Jira server.</p>
<p>FishEye Charts</p>	<p>✓</p>	<p>✓</p>	<p>✓</p>	<p>Shows Lines of Code data from a FishEye repository.</p> <p>If you want to add the FishEye Charts gadget to the dashboard, make sure that:</p> <ol style="list-style-type: none"> 1. The FishEye app is bundled with Jira Software and a working FishEye instance is linked to the Jira instance. 2. A link to the code repository is available. Only three types of repositories are applicable: <code>subversion</code>, <code>perforce</code>, or <code>cvs</code>. <p>Contact the Jira admin about these steps.</p>

FishEye Recent Changesets	✓	✓	✓	<p>Shows two charts about your repository in one: Lines of Code and commit activity.</p> <p>If you want to add the FishEye Recent Changesets gadget to the dashboard, make sure that:</p> <ol style="list-style-type: none">1. The FishEye app is bundled with Jira Software and a working FishEye instance is linked to the Jira instance.2. A link to the code repository is available. Only three types of repositories are applicable: <code>subversion</code>, <code>perforce</code>, or <code>cvs</code>. <p>Contact the Jira admin about these steps.</p>
----------------------------------	---	---	---	---

Using the help center

Your service project comes with an online help center for your customers.

From the help center, customers can browse all the service project portals they have access to. They can do a lot of other things too like:

- View available service projects, and filter them by popularity or alphabetical order.
- Browse a specific service project.
- Search for request types and knowledge base articles.
- See requests they have raised.

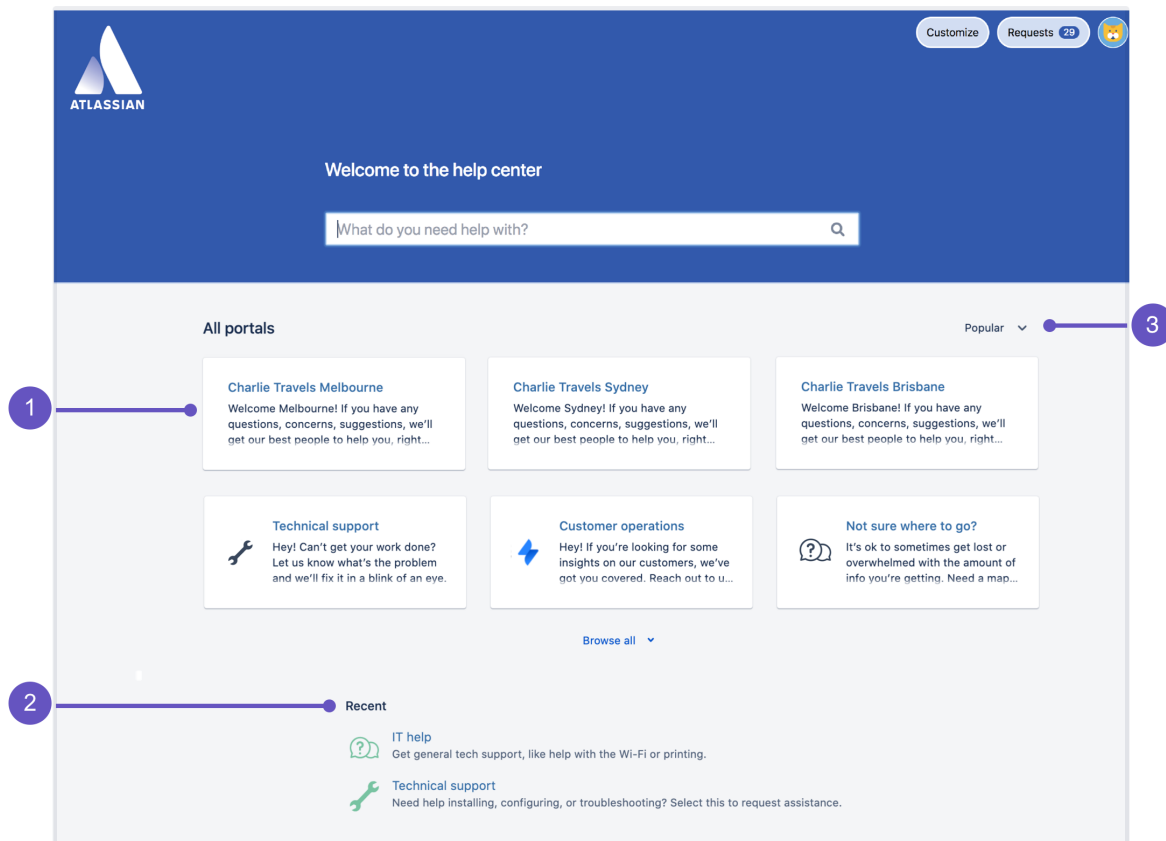
It's good practice to direct your customers to the help center so that they don't have to remember, say, how to submit a request for a new laptop. They simply search for 'new laptop' in the search field and the correct request type will be offered up.

On this page:

- [Customizing the help center](#)
 - [Brand the help center](#)
 - [Set up a knowledge base](#)
 - [Share the help center with customers](#)
- [Manage access to the help center](#)
- [How the smart search works](#)
 - [What you can do to improve search](#)

Customizing the help center

Here's a quick look at the help center layout:



1. **Portal cards:** View all customer portals at a glance and switch between them quickly.

2. **Recent:** View the types of requests that you raised recently.
3. **Sorting:** Sort customer portals by popularity, or the ascending or descending order. The popularity of your portals is based on the number of requests raised.

To help your customers get the most out of your help center, we recommend that you do three things:

- Brand your help center.
- Set up a knowledge base for self-service.
- Share the help center with customers.

Brand the help center

Jira admins, project admins, and agents with permission can brand the help center in a number of ways. Here's how:

1. Go to **Jira administration > Applications > Jira Service Management > Configuration**.
2. Below **Help center**, select the link **View and customize**.
3. In the customize panel on the right hand side do any of the following:
 - Add a help center announcement
 - Add an announcement to the log in screen
 - Give your help center a name
 - Add a logo
 - Add a banner image or banner background color
 - Change the color of components
- Select **Save changes** to publish your announcement.

The screenshot shows the configuration interface for branding the help center. It features a sidebar on the left with a 'Requests' badge showing '30' and a help center icon. The main content area is divided into three sections:

- Add help center announcements:** Includes fields for 'Announcement subject' (with the value 'Hours of operation'), 'Announcement message' (with the value 'Mon-Fri: 09:00 to 17:00' and 'Sat-Sun: 09:00 to 12:30'), and 'Links [name|http://example.com]'.
- Add login announcement:** Includes fields for 'Announcement subject' (with the value 'e.g. login support, signup instructions'), 'Announcement message', and 'Links [name|http://example.com]'.
- Customize help center:** Includes fields for 'Help center name' (with the value 'Help center' and a note '(Also included in breadcrumbs)') and 'Front page title' (with the value 'Welcome to the help center').

At the bottom of the configuration panel, there are two buttons: 'Save changes' and 'Cancel'.

Branding doesn't stop at just the help center. You can also brand the customer portal that belongs to a service project. To learn how to do this, check out [Configuring the customer portal](#).

Set up a knowledge base

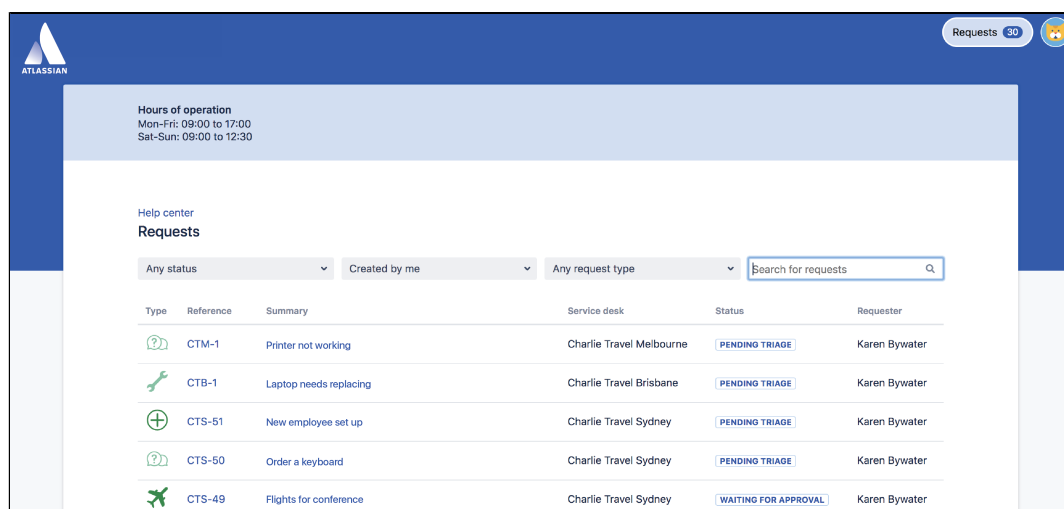
If you have Jira Service Management and Confluence, you can deflect common requests and solve issues faster by setting up a knowledge base. To learn more, check out [Set up a knowledge base for self-service](#).

Customers can read articles in the help center and mark articles as helpful or not. If customers still need to contact your team after reading a knowledge base article, they can select one of the suggested request types or browse all customer portals they have access to.

Share the help center with customers

The help center shows all of the customer portals in your Jira Service Management instance. From here, customers can raise requests in any of the portals they have access to. They can also view all the requests they've raised via the **Request** button in the top right hand corner of the screen.

Check out [View requests in the help center](#) to learn how this works.



To share the help center with customers, send them the following link:

`http://<computer_name_or_IP_address>:<http_port_number>/jira/servicedesk/customer/portals`

You can also add customers to a service project which gives them an account in the help center.

Manage access to the help center

You can choose who can raise requests in your help center and who your customers can share requests with.

To learn more about different access options, check out [Managing access to your service project](#).

i The customer portal integrates with [Atlassian Crowd](#), Atlassian's single sign-on (SSO) framework. For information about integrating with third-party SSO providers check out [Single sign-on integration with Jira and Confluence](#).

How the smart search works

The smart search algorithm learns from past searches and request types raised, so if a customer has previously raised a hardware request for a laptop and monitor they can search for *laptop* or *monitor* in future to find the same hardware request type. It also performs a similarity search, so if you searched for *Onboard employee* you'd see the *Onboard new employee* request type in the results.

Furthermore, the smart search [has been built to be language-agnostic](#), so can learn from search words or phrases entered in any language. As customers enter more searches and raise more requests, the search algorithm gets smarter regardless of the language used.

What you can do to improve search

Your team can help improve search results by updating the request type field when a customer has for example, searched for *new laptop* and raised a software request instead of a hardware request. To learn more, check out [Setting up request types](#).

View requests in the help center

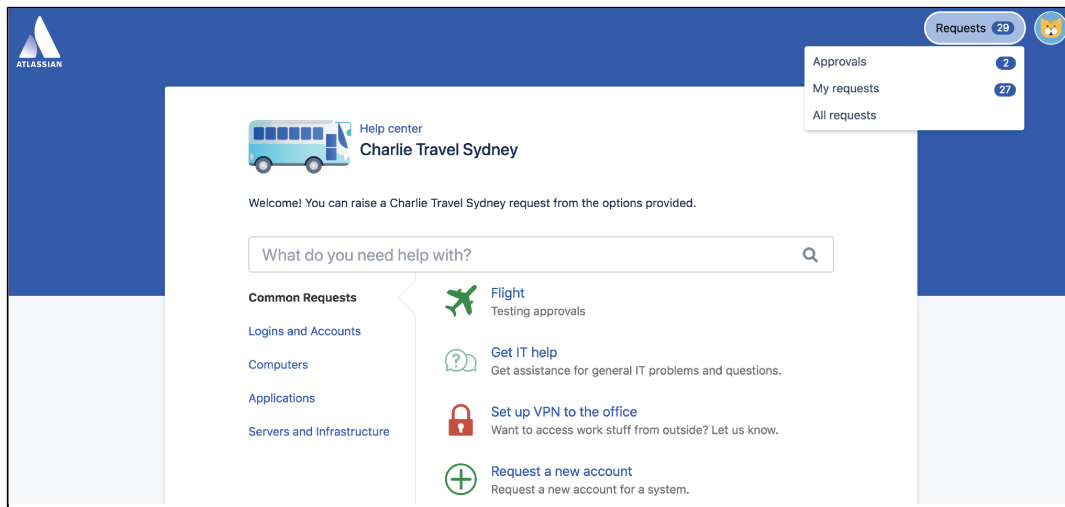
When a customer raises a request in a service project, they receive a confirmation email and can correspond with the agent via email. Alternatively, if customers have an account they can view all their requests via the help center and customer portals they have access to. [Learn more about customer portals](#)

On this page

- [How to view requests](#)
- [How to sort requests](#)
- [How to find the help center](#)

How to view requests

When logged in, you'll see a button in the top right corner that says **Requests**.



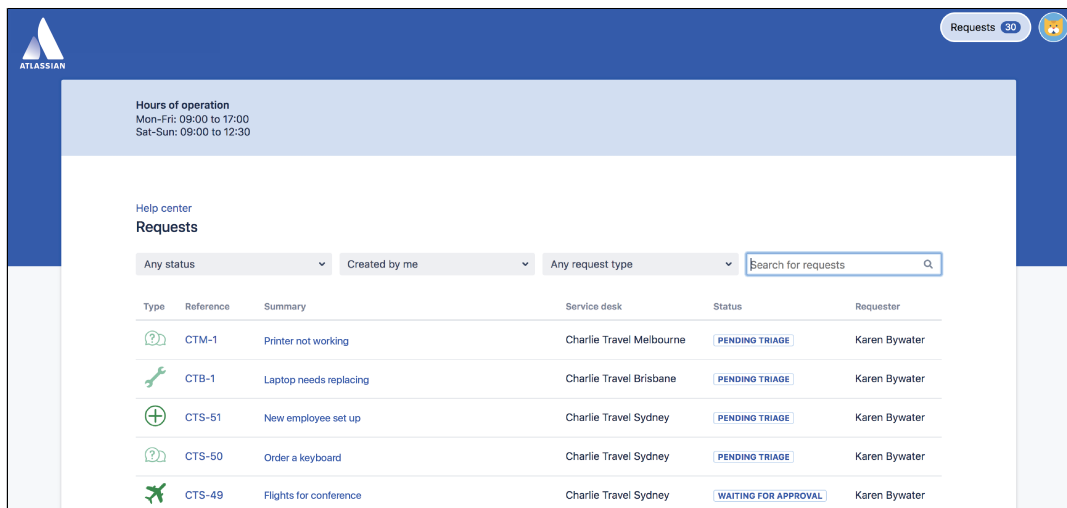
Here, a customer can quickly access the following:

- Approvals - a request submitted by another person that they are an *approver* on.
- My requests - a request submitted by that person.
- All requests - both of the above, as well as a request submitted by another person that they've been added to as a request participant.

How to sort requests

From the requests page a customer can filter their requests by the following parameters:

- Request status - for example, open or closed.
- Request type - for example, software requests or email access.
- Created by - for example, created by me or created by someone else, but shared with me.



Hours of operation
Mon-Fri: 09:00 to 17:00
Sat-Sun: 09:00 to 12:30

Help center
Requests

Any status Created by me Any request type Search for requests

Type	Reference	Summary	Service desk	Status	Requester
	CTM-1	Printer not working	Charlie Travel Melbourne	PENDING TRIAGE	Karen Bywater
	CTB-1	Laptop needs replacing	Charlie Travel Brisbane	PENDING TRIAGE	Karen Bywater
	CTS-51	New employee set up	Charlie Travel Sydney	PENDING TRIAGE	Karen Bywater
	CTS-50	Order a keyboard	Charlie Travel Sydney	PENDING TRIAGE	Karen Bywater
	CTS-49	Flights for conference	Charlie Travel Sydney	WAITING FOR APPROVAL	Karen Bywater

A request will display on this page only when a customer has created that request, or if they have been added as a participant on someone else's request. Learn more at [Adding request participants](#).

How to find the help center

To share the help center with customers, you can link to the following URL:


```
http://<computer_name_or_IP_address>:<http_port_number>/jira/servicedesk/customer/portals
```

You can also add customers to a service project which gives them an account in the help center. Learn more at [Adding customers](#).

Configuring request view

Your customers with accounts can view all their requests in the help center. Its up to you which columns are available to them, though. Selecting specific columns will allow your customers to see and configure those columns in their requests list.

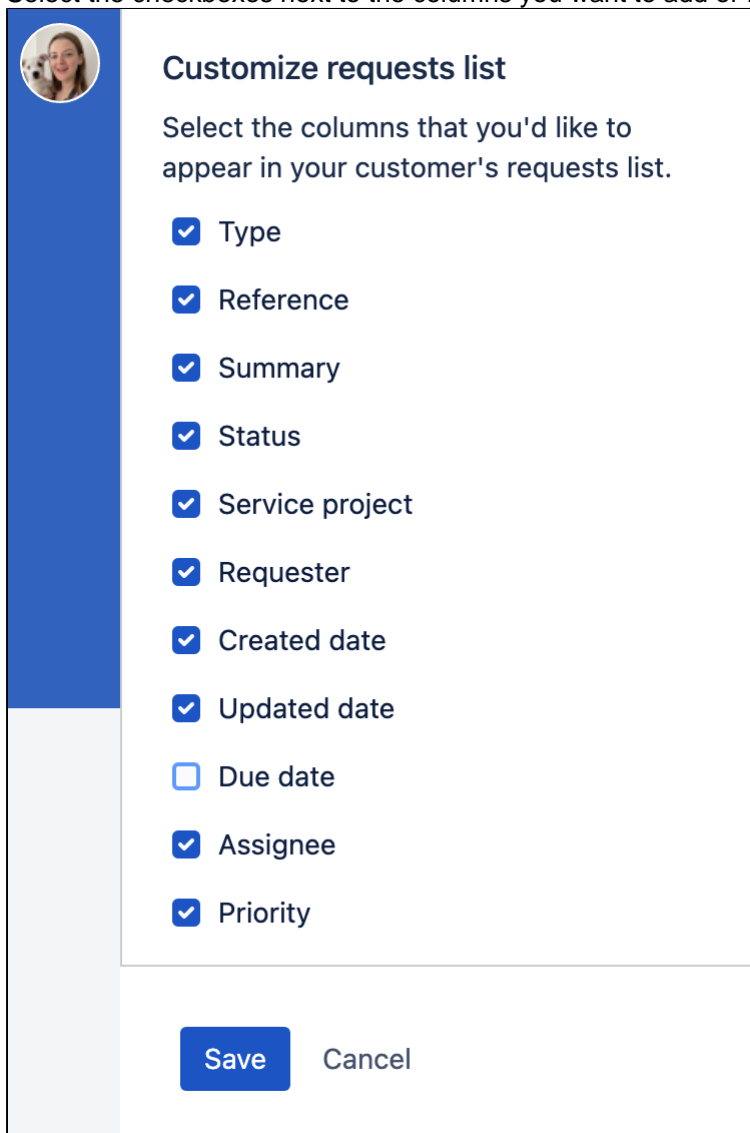
Customize available columns as an admin

 You need to be a Jira admin to customize the request view.

 Once you save this configuration, it applies globally to all requests.

To select columns available to your customers:

1. Go to the help center.
2. Select **Requests > All requests**.
3. From the top navigation bar select **Customize**.
4. Select the checkboxes next to the columns you want to add or remove.



Customize requests list

Select the columns that you'd like to appear in your customer's requests list.

- Type
- Reference
- Summary
- Status
- Service project
- Requester
- Created date
- Updated date
- Due date
- Assignee
- Priority

Save Cancel

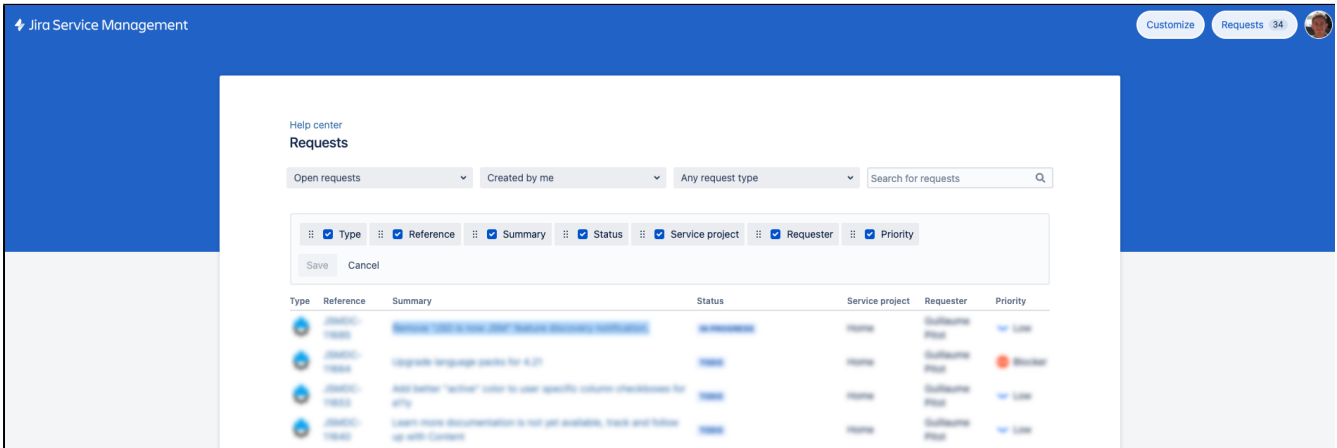
5. Select **Save**.

Configure the requests list view as a customer

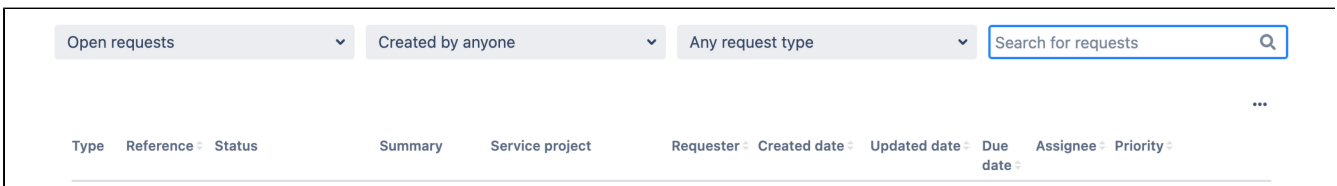
After you specify which columns are available to your customers, they can configure their requests list view according to their needs. Customers can change:

- Which columns they want to see in their requests list
- The order of columns
- How columns are sorted

To configure the requests list view, a customer needs to go to the help center, then select **Requests > My requests**. Selecting the **more actions** menu (⋮) will enable them to configure the columns in their requests list and their order.




To sort a column, the customer needs to select the column title.



You can sort the following columns:

Column name	Sorting type
Reference ⓘ Also known as Issue key.	<ul style="list-style-type: none"> • Sorted A > Z • Sorted Z > A
Requester	<ul style="list-style-type: none"> • Sorted A > Z • Sorted Z > A
Created date	<ul style="list-style-type: none"> • Sorted oldest > newest • Sorted newest > oldest
Updated date	<ul style="list-style-type: none"> • Sorted oldest > newest • Sorted newest > oldest

Due date	<ul style="list-style-type: none">• Sorted oldest > newest• Sorted newest > oldest
Assignee	<ul style="list-style-type: none">• Sorted A > Z• Sorted Z > A
Priority	<ul style="list-style-type: none">• Sorted in ascending order• Sorted in descending order <div data-bbox="571 566 1286 647" style="border: 1px solid #ccc; border-radius: 5px; padding: 5px;"> Descending order = highest priority > lowest priority.</div>

Organizing work with components

Components are used to organize or group customer requests in a service desk project. You could set up a component for systems that your teams are responsible for (e.g. company intranet), and then set a default assignee so that any customer request about that system is assigned to the agent who manages it.

You need to have the project-specific **Administer Projects** permission or the **Jira Administrator** global permission to be able to:

- Add create a new component against which issues can be aligned
- Edit change a components details
- Delete remove a component

Once a component has been created for a project, the 'Component' field becomes available for your issues. If you cannot see this field on your issue, your project may not have any components yet, or the field is hidden from view.

Managing a project's components

The easiest way to manage a project's components is through the Components page.

1. Choose **Administration**  > **Projects**, and click the name of the project.
2. Choose **Components** in the sidebar. The **Components** page is displayed, showing a list of components and each component's details. From here, you can manage the project's components as described below.

Adding a new component

1. The Add Component form is located at the top of the 'Components' screen.
2. Enter the **Name** for the component. Optionally, enter a **Description** and select a **Component Lead** and **Default Assignee** (see [options](#) below).
3. Click **Add**.

Selecting a default assignee

You can optionally set a default assignee for a component. This will override the project's default assignee for issues in that component. If an issue has multiple components, and the default assignees of components clash, the assignee will be set to the default assignee of the component that is first alphabetically.

Default assignee option	Description	Notes
Project Default	Issues matching this component will have the assignee set to the same default assignee as the parent project.	
Project Lead	The assignee will be set to the project leader.	If the project leader is not permitted to be assigned to issues in the permission scheme, this option will be disabled and will say "Project Lead is not allowed to be assigned issues".

On this page:

- [Managing a project's components](#)
- [Adding a new component](#)
- [Selecting a default assignee](#)
- [Editing a component's details](#)
- [Deleting a component](#)

Component Lead	The assignee will be set to the component leader.	If the component leader is not permitted to be assigned to issues in the permission scheme, this option will be disabled and will say "Component Lead is not allowed to be assigned issues". The Component Lead option will also not be available if the component does not have a lead assigned to the component. Instead, under this option, it will say "Component does not have a lead".
Unassigned	The assignee of the issue will not be set on the creation of this issue.	This option will only be available if "Allow unassigned issues" is enabled in the general configuration.

Editing a component's details

1. On the 'Components' screen, open the menu in the **Actions** column for the component you want to edit, and select **Edit**.
2. Edit the component's **Name**, **Description**, **Lead**, and **Default Assignee** in the **Edit component** dialog.
3. Click the **Save** button to save your changes.

Searching for a component

If you need to find a component in a long list, it's easiest to search for it. Start typing text into the search box that you know the component contains, and your list will automatically be filtered for you.

Deleting a component

1. On the 'Components' screen, open the menu in the **Actions** column for the component you want to delete, and select **Delete**.
2. You will be prompted to associate any issues assigned to this component with another component if you wish.
3. Click the **Delete** button to delete the component.

Set up a knowledge base for self-service

Linking Jira Service Management to Confluence and using it as a knowledge base lets your agents share their expertise through knowledge base articles and how-tos. They can be linked from requests, searched in Confluence, or shared via direct links. Customers can then help themselves by finding the right solutions, and new agents can quickly get up to speed with the most common problems.

Confluence Cloud



Choose this option if you're using Confluence Cloud, managed by Atlassian.

[Learn more](#)


Confluence Server



Choose this option if you're using on-premise Confluence, managed by your organization.

[Learn more](#)

Set up a knowledge base with Confluence Server

 Follow this step by step guide for linking Jira Service Management to a Confluence knowledge base, so that customers can help themselves and agents can share their expertise.

- [1. Install Confluence](#)
- [2. Get the right number of Confluence licenses](#)
- [3. Link Jira Service Management to Confluence](#)
- [4. Learn about knowledge base settings and permissions](#)
- [5. Link your project to a Confluence space](#)
- [6. Write and search for knowledge base articles](#)
- [Known issues](#)



1. Install Confluence

User:

You'll need **Confluence 5.10 and above** and **Jira Service Management 3.1 (formerly known as Jira Service Desk) and above** to access knowledge base features. Get the latest versions of each for all the bells and whistles.

Steps

1. [Install](#) the latest version of Confluence.
2. Have **thesame user base** in both Jira and Confluence by one of the following methods:
 - [Connect to Crowd or Jira](#) for user management
 - [Connect to an internal Directory](#) with authentication
 - [Connect to an directory](#)
3. Check that you're an **administrator** in both Jira and Confluence.

Good to know

- Read [Use Jira applications and Confluence together](#) to check version compatibility between products.



Having the same user base in both Jira and Confluence means you won't have to manually create and maintain customer accounts.



2. Get the right number of Confluence licenses

User:

Your **service project agents will need a Confluence license** to create and edit knowledge base articles. However, service project customers won't need a Confluence license to view articles.

Good to know

- Read [Confluence licensing and pricing](#) to learn more.




3. Link Jira Service Management to Confluence

User:

Linking two applications allows you to share information and access one application's functions and resources from within the other.

Steps

1. Go to **Jira Administration**  **Applications > Application links**.
2. Enter the URL of your Confluence site and select **Create new link**.
3. Check **The servers have the same set of users**, to configure using OAuth (with impersonation) authentication.
4. If you're *not* an admin on both servers you won't be able to set up a 2-way (reciprocal) application link. If you want to go ahead and create a 1-way link anyway, clear the **I am an administrator on both instances** checkbox.
5. Use the wizard to finish configuring the link.

Good to know

- Read [Linking to another application](#) to learn how to set up a reciprocal link from Confluence.
- Read this [troubleshooting article](#) if you get into difficulty linking Jira to Confluence.



4. Learn about knowledge base settings and permissions

The permissions you set in Confluence determine whether (or not) your agents can create articles and if your customers can view them.

Read [Knowledge base settings and permissions](#) to decide what you need.




5. Link your project to a Confluence space

User: PROJECT ADMINS

When you link a Confluence space to your service desk project, agents can search for solutions and create new articles for common requests. Customers can then use the articles to self-service problems.


There are two ways to link your service desk to a Confluence Space:

Link to an existing space

1. Go to **Knowledge base**  > **Link existing space**.
2. Select the space you want from the dropdown.
3. Choose the link **Define who can view knowledge base articles** to set permissions.

You can also unlink or change spaces in **Project settings**  > **Knowledge base**.

Link to a new space

1. Check you have *Confluence admin* and *Create space* permissions.
2. Go to **Knowledge base** .
3. Select **Create new space**.
4. Congratulations, you have created a new Confluence space.

Good to know

- Read [Use Confluence as a knowledge base](#) to learn how to set up a knowledge base blueprint.



6. Write and search for knowledge base articles

User: AGENTS

Once you've linked to a Confluence space, agents are ready to start writing and sharing knowledge with customers and teammates.

Read [Write and search for knowledge base articles](#) to learn how to do this.



Known issues

If you run into problems connecting a Confluence knowledge base and Jira Service Management, check out [this page](#). It lists common integration-related issues and explains how to troubleshoot them.

Knowledge base settings and permissions

Now that you've linked your Jira Service Management project to a Confluence space, you'll need to set your agents and customers up with the right permissions.

On this page:


- [Who needs a Confluence license](#)
- [Choose who can view articles](#)
- [Check if agents can create articles](#)

Who needs a Confluence license

If you want your service project agents to create, comment on, and search the spaces on your Confluence site, they'll need a Confluence license. Your service project customers however, don't need a Confluence license to view knowledge base articles.

There are three levels of permissions in Confluence that determine who can view and create articles:

- [Global permissions](#) are **site-wide** and can be assigned by a system or Confluence admin.
- [Space permissions](#) are **space specific** and are managed by the space admin.
- [Page restrictions](#) affect both **viewing and editing** and are managed by space admin.

 If you, your agents or your customers are having difficulty viewing articles, check the permissions with your Confluence admin.

Choose who can view articles

Go to **Project settings**  > **Knowledge base** > **Access** > **Viewing**.

Access

Viewing

Define who can view knowledge base articles through the portal and in the linked Confluence space.

- All active users and customers can access the knowledge base without a Confluence license.
- Only licensed users who have [access to the space](#)

This setting determines who can view articles via the help center or a link your team shares. You have two options to choose from:

Option 1. All active users and customers

Users who don't have a Confluence license can view knowledge base articles via the customer portal and help center.

Choose this if:

- You want your team to write articles and share them with customers.

Limitations:

- If your Jira Service Management site and Confluence site have separate user bases, you'll need to create a Confluence user account for each service project customer. If you don't want the customer to use a Confluence license, don't assign the Confluence user to a group.

Option 2. Only licensed users

Users who don't have a Confluence license can't read knowledge base articles unless you allow anonymous access in the knowledge base space.

Choose this if:

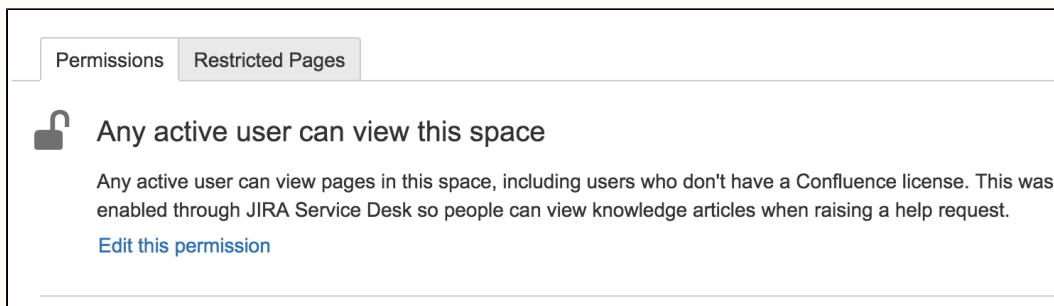
- You only want to use your knowledge base for internal articles.

Limitations:

- Anonymous access is not compatible with SSO using 2-legged OAuth.

Data privacy

If you choose the **All active users and customers setting**, you'll see the following message in the Confluence space permission screen:



This permission overrides all existing space permissions. *Any logged in Confluence user* will be able to see the space (regardless of their group membership).

You can disable this permission at any time, but it can only be re-enabled from Jira Service Management.


Login process

If you aren't using SSO or something that passes login information between instances, then users will need to log in to Confluence using their *Jira Service Management credentials* to view articles.


Check if agents can create articles

Your agents will need a Confluence license to create and edit articles, and permission to create articles in the Confluence space.

To check if an agent can create articles:

1. Go to **Project settings**  > **Knowledge base** > **Access** > **Authoring**.
2. Select the **space permissions** link, this will take you to your space permission's page.
3. Check that the agent (or a group they are a member of) has the **Add page** permission.

Set up a knowledge base with Confluence Cloud

 Follow this step by step guide for linking Jira Service Management to a Confluence Cloud knowledge base so that customers can help themselves and agents can share their expertise.



1. Set up a Confluence Cloud account

You'll need a Confluence Cloud account to create the integration. If you don't have one already, check out this link:

- [Try Confluence Cloud](#)



2. Get the right number of Confluence licenses

Your service project agents will need a Confluence license to create and edit knowledge base articles. However, service project customers won't need a license to view these articles.

Good to know

- Read [Confluence Cloud licensing and pricing](#) to learn more.



3. Create a user with the right permissions to access Confluence

When integrating Jira Service Management with Confluence, you'll need to choose a user that will be used to access Confluence. **Its important** that this user's permissions are similar to what you'd like customers to have.

Why are permissions important?

When customers **search** your knowledge base, they get results based on the permissions of the user you've chosen. If that's an admin or someone with high permissions, customers might see results they shouldn't have access to.

This will include the title and an excerpt of an article (that's what's displayed in the search). If an article is restricted, they won't be able to view it as the article's restrictions are checking permissions of a specific user who's trying to access it. Nevertheless, it's easy to reveal more than you'd like to, so make sure to set the permissions right.



4. Integrate Jira Service Management with Confluence

Audience: JIRA ADMINS

Once you've covered the requirements on the Confluence side, it's time to integrate it with Jira Service Management. That's a one-time action after which your project admins will be able to link their projects to spaces in Confluence.

Steps

1. Go to **Administration > Applications > Confluence Cloud integration**.
2. Enter the URL of your Confluence site, including the `/wiki` suffix, for example:

```
https://confluence.atlassian.net/wiki
```

3. Enter a descriptive display name. This is what project admins will see when linking their projects to Confluence.
4. Specify the user that will be used to access Confluence. Make sure to set the right permissions for this user, as described in the previous step.
5. Enter your Confluence API token. You can generate it from your Atlassian account. For more info, see [API tokens](#).
6. Click **Save**.

Good to know

- In the past, you might have created integrations with other applications through Application Links. Although this would still work for Confluence Cloud, using it as the knowledge base requires that you create the integration through the **Confluence Cloud integration** page. No AppLinks this time.



5. Learn about knowledge base settings and permissions

The permissions you set in Confluence determine whether (or not) your agents can create articles and if your customers can view them.

- Read [Knowledge base settings and permissions in Confluence Cloud](#) to decide what you need.



6. Link your project to a Confluence space

Audience: PROJECT ADMINS

When you link your service project to a Confluence space, agents can search for solutions and create new articles for common requests. Customers can then use the articles to self-service problems.

Before you begin

- You need to create a space for your project in Confluence beforehand, as you can only link projects to existing spaces.

Steps

- Go to **Knowledge base**>**Link existing space**.
- Select the space you want from the dropdown.
- Choose the link **Define who can view knowledge base articles** to set permissions.

You can also unlink or change spaces in **Project settings**>**Knowledge base**.



7. Write and search for knowledge base articles

Audience: AGENTS

Once you've linked to a Confluence space, agents are ready to start writing and sharing knowledge with customers and teammates.

- Read [Write and search for knowledge base articles](#) to learn how to do this.



Known issues

The pagination of search results is not available in Confluence Cloud, that's why when you search for knowledge base articles, you'll only see the top 3 articles.


When linking a project to a Confluence space, you'll need to create this space in Confluence beforehand. You won't be able to create it from within Jira Service Management.

You also won't be able to create a Confluence space when creating a service desk project.

To view articles linked in your requests, your customers will need to be logged in to their Atlassian cloud accounts (no license required) unless you allow anonymous access. That's a known issue that makes this integration different from using Confluence Server or Data Center.

To check what your customers need to view the articles, how to allow anonymous access for your Confluence spaces, and how to create Atlassian cloud accounts without them consuming extra licenses, see [Knowledge base settings and permissions in Confluence Cloud](#).

There is a feature request to improve this behavior:

 **JSDSERVER-7175** - Knowledge Base articles can't be opened in the customer portal if Confluence Cloud is not configured with anonymous access **GATHERING INTEREST**

Knowledge base settings and permissions in Confluence Cloud

Now that you've linked your Jira Service Desk project to a Confluence Cloud space, you'll need to set your agents and customers up with the right permissions.

Choose who can view articles

First you should decide whether the space linked to your project should be available to all active users and customers (requiring Atlassian cloud account, but not license) or only licensed users, for internal purposes. You can then additionally allow anonymous access for your space.

To configure permissions for your knowledge base:

1. Open your project.
2. Go to **Project settings > Knowledge base**, and select one of the options.

Access

Viewing


Define who can view knowledge base articles through the portal and in the linked Confluence space.

All active users and customers can access the knowledge base without a Confluence license.

Only licensed users who have access to the space

Option 1. All active users and customers

Users need to log in to their Atlassian cloud accounts to view articles, but they don't need product access to your Confluence site, and hence don't consume a license. If you don't want users to log in, you can allow anonymous access.

 To enable this setting, the user you've chosen when integrating with Confluence Cloud needs admin permissions for your space.

Tasks in Confluence	What your users needs
<ul style="list-style-type: none">• Create articles• Comment on articles• Search the spaces	<ul style="list-style-type: none">• Atlassian cloud account• Product access to your Confluence site (consuming a license)
<ul style="list-style-type: none">• View articles linked in requests	<ul style="list-style-type: none">• Atlassian cloud account• Only site access (not consuming a license) <p>With anonymous access allowed:</p> <ul style="list-style-type: none">• Nothing


Option 2. Only licensed users

Users need Atlassian cloud accounts and product access for your Confluence site. Choose this option if you're planning to use your knowledge base for internal purposes and already have licenses assigned to your users. Here you can also enable anonymous access to some Confluence spaces so users don't have to log in to view them.

Tasks in Confluence	What your users needs
<ul style="list-style-type: none"> • Create articles • Comment on articles • Search the spaces 	<ul style="list-style-type: none"> • Atlassian cloud account • Product access to your Confluence site (consuming a license)
<ul style="list-style-type: none"> • View articles linked in requests 	<ul style="list-style-type: none"> • Atlassian cloud account • Product access to your Confluence site (consuming a license) <p>With anonymous access allowed:</p> <ul style="list-style-type: none"> • Nothing

Configuring anonymous access so customers can view articles without logging in

Even if you give access to your knowledge base to all active users and customers (Option 1 above), they still need to be logged in to their Atlassian cloud accounts to view articles linked in your requests. To avoid creating accounts for your customers and making them log in every time, you can allow anonymous access for your Confluence site and chosen spaces.

 Confluence spaces for which you allow anonymous access will be available to anyone on the web.

Configure anonymous access

You need to allow anonymous access for **both your site and space**:

- To allow it for your site, see [Set up public access](#). Note that even with anonymous access allowed for the site, your spaces won't be open to public until you specifically allow that.
- To allow it for your space, see [Make a space public](#).

Creating Atlassian cloud accounts for customers without consuming licenses

If you don't want to enable anonymous access, you can invite your customers to Atlassian cloud so they can log in and view knowledge base articles without consuming licenses. To put it shortly, you'll need to make sure their accounts don't have **product access** enabled for Confluence Cloud.

To disable product access for new users:

1. Go to your Atlassian cloud sites [Admin at admin.atlassian.com](#).
2. In the sidebar, select **Product access**.
3. Clear the **New users have access to this product** option next to your Confluence site.
4. When inviting new users, make sure that your Confluence site isn't selected in **Product access**, and that users aren't added to any groups that have product access (e.g. confluence-users).

To check the product access for existing users:

1. Go to your Atlassian cloud sites [Admin at admin.atlassian.com](#).
2. In the sidebar, select **Users**.
3. Check the details of your users:
 - a. Make sure they don't have access granted for Confluence. They should only have the **Has access on site** option enabled.

- b. Make sure they aren't added to any groups that have product access (e.g. confluence-users).

Write and search for knowledge base articles

Your knowledge base is there for customers, even when you're working on other requests, at home for the day, or away on holiday.

Here are some reasons you might write knowledge base articles:

- **You get a lot of similar requests.** Like "How do I access office wifi?". When customers search for 'wifi' in the help center, they'll find your article. If they send you a request, you can share the article rather than walking them through the steps in a comment.
- **You're upgrading a system.** You can write a step by step upgrade guide and link to it from an announcement in the help center.
- **You're troubleshooting the same issue with customers.** If you normally walk people through a series of steps to diagnose and troubleshoot a problem, such as a broken printer, a troubleshooting guide will save you time.

On this page:

- [How to write articles](#)
- [How to share articles](#)
- [How the search works](#)
- [How to access search help](#)

How to write articles

After you set up your knowledge base space, the issue view displays a panel called **Related knowledge base articles**:

AD-21

How do I connect to the file server?

Edit Comment Assign More Resolve this issue Respond to customer

Details

Type: Service Request Status: **WAITING FOR SUPPORT**
(View Workflow)

Component/s: None Resolution: Unresolved

Labels: None

Description

I am trying to access some documents on the file server and cannot seem to be able to connect to

Attachments

Drop files to attach, or browse.

Related knowledge base articles

[I think that the intranet might be down. What do I...](#) 🔒

Didn't find a suitable article? [Search knowledge base](#) or [create an article](#).

Activity

All Comments Work Log History Activity

If you don't see a relevant article, you can create a new one from the issue. You'll need the **Add page** permission in the Confluence space. Customers can read any article that doesn't have a red padlock.

Related knowledge base articles

[I think that the intranet might be down. What do I...](#) 🔒

Didn't find a suitable article? [Search knowledge base](#) or [create an article](#).

Activity

All Comments Work Log History Activity

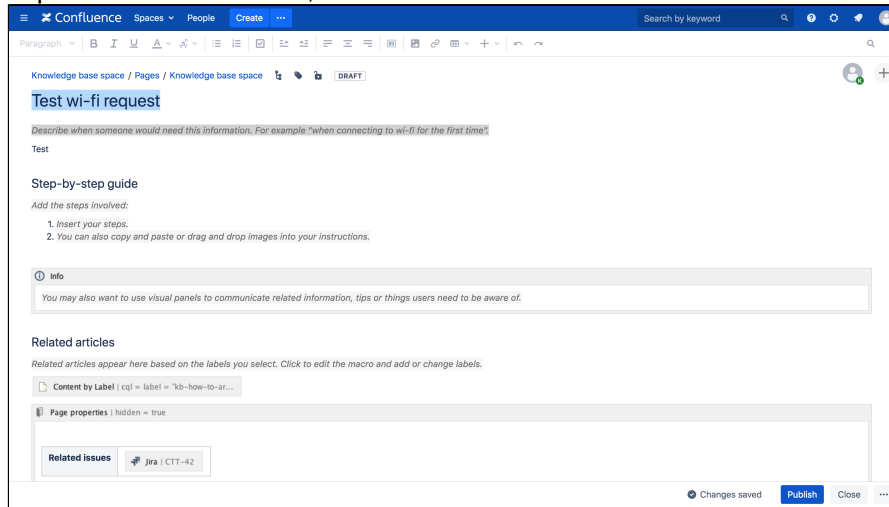
How do I connect to the file server?

How-To

Create Cancel


To create a new article:

1. Click the **create an article** link.
2. By default, the issue summary becomes the new article's title, you can edit this.
3. Choose either the **How-To** or **Troubleshooting** template.
4. Follow the prompts to create the article, then **Publish**.



5. Check how it displays in the issue view.

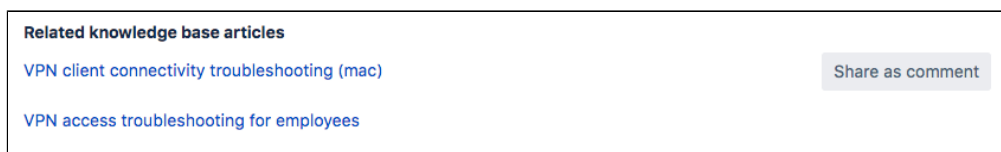
i Page labels are essential in knowledge base spaces. These are used to add topics to your articles, and allows your knowledge base to become self-organizing over time.


You can also create articles from the sidebar, then **Knowledge base**  > **Create article**.

How to share articles

There are two ways you can share articles with customers:

- Hover over an article to share it as a comment on a request:



- Share links from a knowledge base article that you access via **Knowledge base**  in the sidebar.

When customers click the link from a request or their email, the article opens in the help center:

Teams in Space Help Cen...

How to Connect to the WiFi Network from a PC

This guide will help you to understand how to connect your computer to wifi. It's basic procedure is very similar in other versions of Windows and in Mac OS X.

Follow these step-by-step instructions to connect to wifi

Step 1: Set up your wireless router - an example of which is on the right - (see How instructions). Most internet providers now supply wireless routers as standard. What's important to provide appropriate security so that your computer can't be entered by for this should be supplied with the router, but if in doubt, consult an expert.

Step 2: Check that your computer has a built-in wireless adaptor (see left). Up-to-date one, but most desktop computers don't.

To check whether there's a built-in adaptor, follow these steps:

How the search works

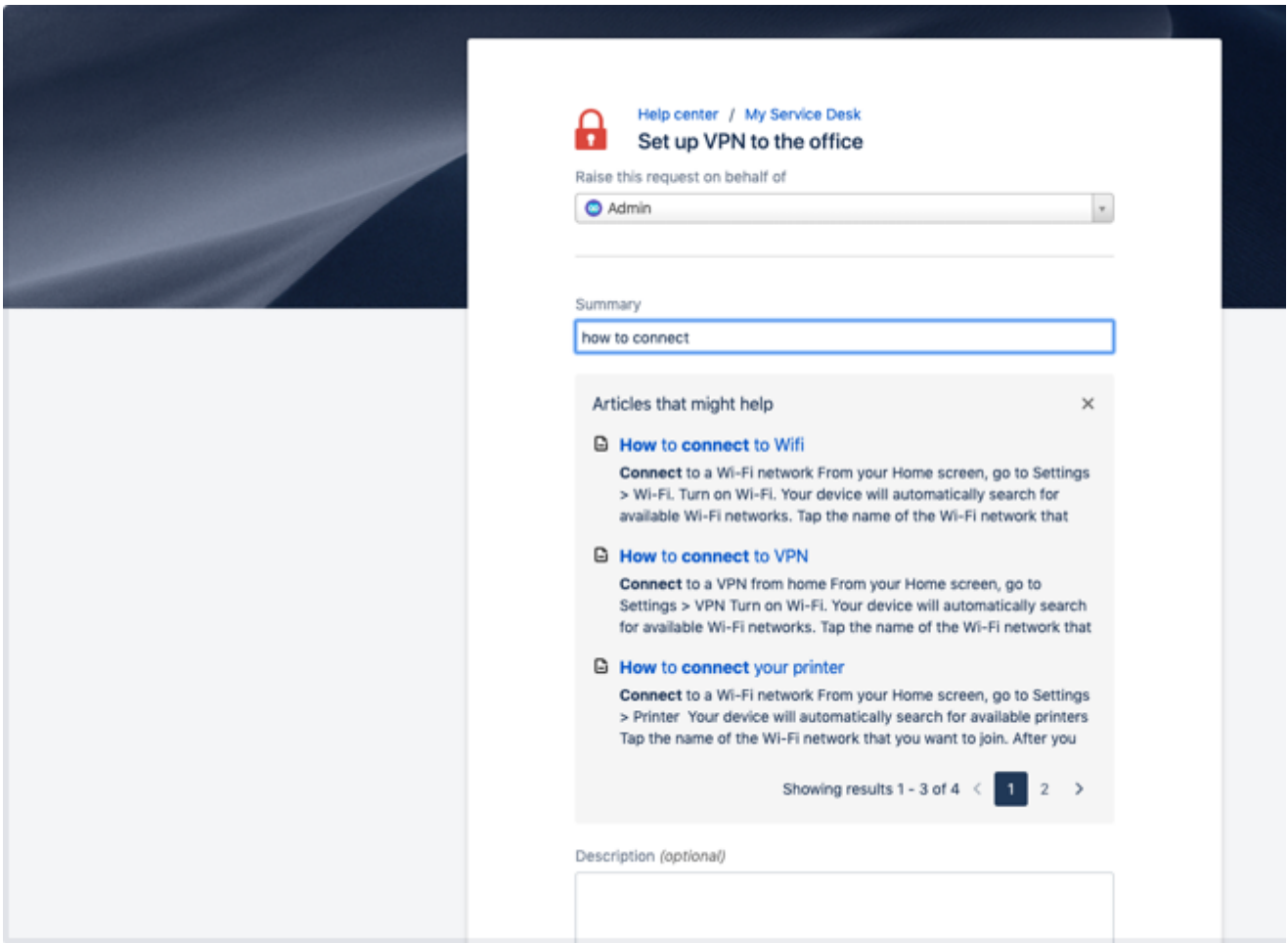
The knowledge base recommends articles to customers and agents using keywords it pulls from the **Summary** field of a request.

Help center and customer portal search results

When a customer types something in the search field on the help center or customer portal, the results return the following:

- Related knowledge base articles. These help customers resolve their own issues, without needing to raise a request.
- Related request types. These are recommended to customers, so they know which request type to fill out based on their issue.

To make it even easier for customers to find what they need, you can recommend articles when they fill out a particular request type.



Recommending articles

To recommend articles for certain request types, go to **Project settings > Knowledge base > Auto-search on request types**, then in the **Search KB** column, choose **Yes**:

Request form	Search KB	Restrict to articles with labels
Technical support	Yes	<input type="text" value="Technical_support"/>

You can restrict the search to articles with a certain label. To make life easier for yourself, use a label similar to the name of the request form.

⚠ You can't use labels as a way of restricting access to an article. For example, if you create a knowledge base article and add the label "display", and then create another article and don't add the label "display" the page without the label will still display in the search. Adding labels are a way to filter related articles, not assign viewing permissions.

If you need to restrict an article so that only certain users can view it, do this through [knowledge base settings and permissions](#).

Using multiple knowledge bases

If you have multiple knowledge bases (Confluence instances) connected to your Jira Service Management instance and are searching through the Help Center, only articles from the primary knowledge base will be returned. This is a known issue related to the Help Center, you can follow a feature request [here](#). Articles from remaining knowledge bases will be returned when searching on individual customer portals.

You can set any Confluence instance as the primary one by going to **Administration > Applications > Application links**.

Other important information

- **Knowledge base vs. request type search:**The knowledge base search and the request type search are independent of each other.
- **Request type search:**The request type search will display a maximum of 5 results, and:
 - The primary search picks up words that the customer is typing in the **Summary** field, and looks for them in the request type's **Name** and **Description**.
 - The secondary search uses JQL queries to find relevant request types based on the request's **Summary** field.

Issue view search results

When agents view a request, the knowledge base recommends related articles that they can reference or share with customers. Your service project will search the Confluence space you have linked for any articles that contain keywords in the request's **Summary** field.

Important things to note:

- The search ignores all other fields from the request (only the summary field is used).
- Check that your request types have a visible **Summary** field. Read [Setting up request types](#).
- The search skips articles that the agent doesn't have permission to view.
- The search and ranking follows the same rules as the [Confluence search](#). In short, Confluence identifies all articles that contain keywords matching those in the request's **Summary** field, and sorts them by word frequency.
- Agents can only share articles that the customer has permission to see.

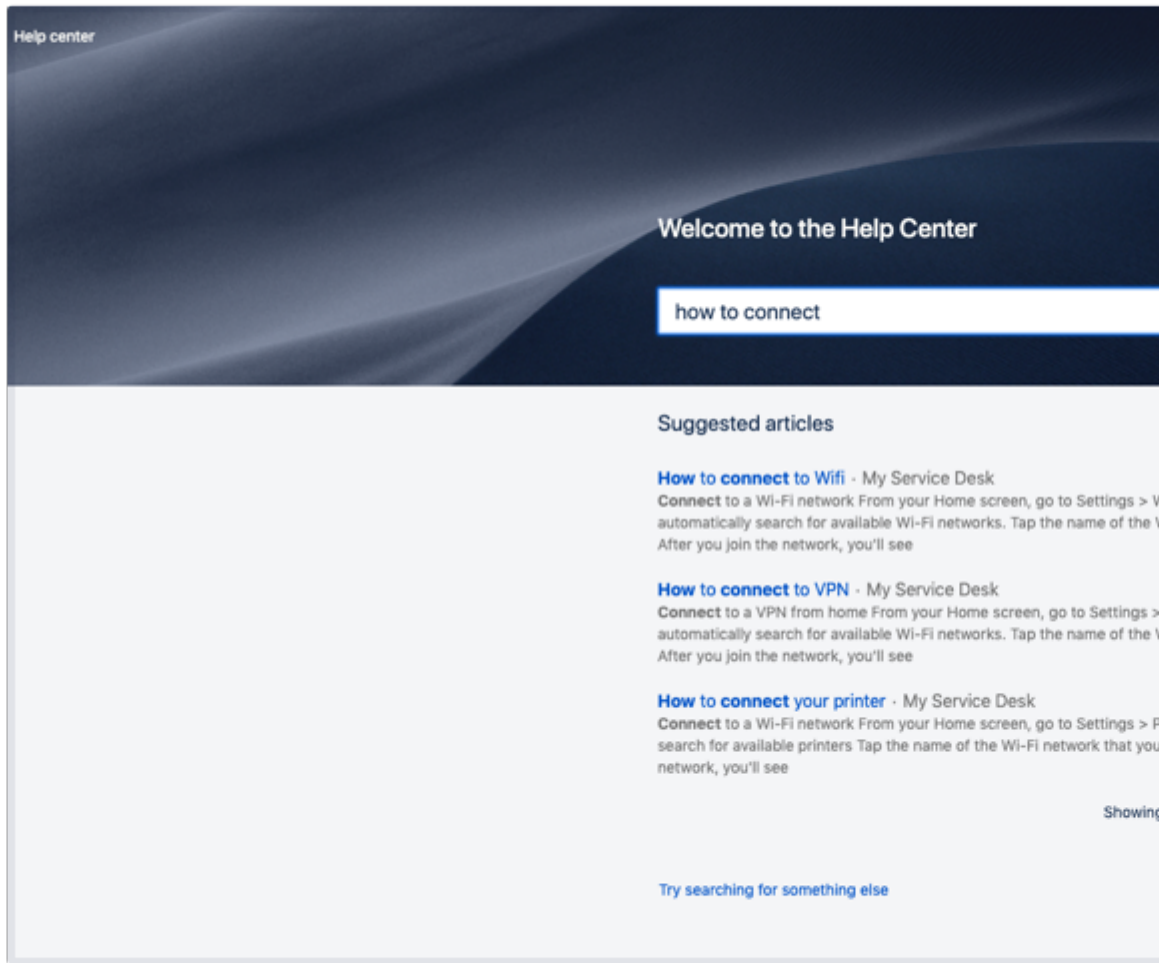


Change the article title to include more keywords from the request summary field, if you don't see an article that you think you should.

Agents can also search for articles via the **Knowledge base** in the project sidebar.

How to access search help

When you link a Confluence knowledge base to your service project, a **Search help** link appears below the search field.



When you click on this link, you're shown a screen that explains how to create an advanced search query using Confluence search syntax (including wildcards) for more precise knowledge base search queries.

Search the knowledge base

Exact phrase search
To search for content that contains the exact phrase "chalk and cheese":
`"chalk and cheese"`

OR search
To search for content that contains one of the terms, "chalk" OR "cheese":
`chalk OR cheese`

AND search
To search for content that contains both the terms "chalk" AND "cheese":
`chalk AND cheese`

NOT search
To search for content that contains "chalk" but NOT "cheese":
`chalk NOT cheese`

Excluded term search
Similar to the NOT search, to search for content that contains "chalk" and "butter" but NOT "cheese":
`chalk butter -cheese`

Close

Check out [Confluence Search Syntax](#) for an overview of this function.

Setting up incident management

You can set up incident management to allow your agents to view and manage Opsgenie incidents within their Jira Service Management projects. This allows them to quickly respond to disruptions, get up to speed with whats going on, create incidents and provide more context for the Opsgenie teams to resolve them.



1. Set up your Opsgenie account

Major incidents are stored and managed in Opsgenie, an Atlassian application used for alerts. You'll need an Opsgenie account to integrate it with Jira Service Management. If you don't have one already, check out this link:

- [Try Opsgenie](#)



2. Integrate Jira Service Management with Opsgenie

User: JIRA ADMIN

Integrating Jira Service Management with Opsgenie is a one-time action that will let the two applications exchange information between one another. Once that's done, every project admin can enable incident management for each project separately.

Before you begin

- You'll need the Opsgenie API URL and API key to create the integration. Check [Getting Opsgenie API information](#) to learn how to find them.

Steps

1. Go to **Administration > Applications > Opsgenie integration**.
2. Enter your Opsgenie API URL and API key.
3. Select **Save**. Your integration is now enabled.



3. Enable incident management for your projects

User: PROJECT ADMIN

If you'd like to use incident management in your project, you can enable it in project settings. This will give agents working on a project access to ongoing major incidents, and some special permissions like linking incident to requests.

1. Go to **Projects settings > Incident management**.
2. Allow agents to view, link, and manage major incidents within this project.



4. Work with ongoing major incidents

User: AGENT

Enabling incident management brings a new screen to your Jira Service Management where you can view ongoing major incidents to be up to date with any disruptions, and link them to requests to help your Opsgenie teams resolve them faster.

4

Ongoing major incidents

Opsgenie is showing these major disruptions (incidents) right now.

All incidents

Open

Past

P	Key	Message	Opsgenie service/s	Responders	Status	Linked	Created
P2	#52	Space shuttle won't start	third service	Second team	OPEN	1 request	19/Jul/20 11:47 PM
P3	#51	Problem with rocket thrusters		Confluence Team	OPEN	5 requests	17/Jul/20 5:45 AM
P3	#50	Someone stole the Moon	second service	Test Team	OPEN	5 requests	16/Jul/20 3:11 AM
P3	#49	Robot overboard!	second service	Test Team	OPEN	3 requests	16/Jul/20 3:07 AM
P2	#48	Wormhole is down	Trading Service	Athena	OPEN	0 requests	16/Jul/20 1:49 AM
P4	#47	Where's my ship	Translations Service	Translations Team	OPEN	3 requests	16/Jul/20 12:38 AM
P3	#46	Refunds system is down	Translations Service	Translations Team	OPEN	0 requests	15/Jul/20 1:28 AM
P3	#45	Oxygen depletion		Jira Team	OPEN	3 requests	15/Jul/20 1:25 AM
P3	#44	Mars office is dusty	Confluence Service	Confluence Team	OPEN	3 requests	15/Jul/20 1:19 AM
P1	#43	Incident Management has problems	Sample Service	Test Team	OPEN	5 requests	14/Jul/20 12:30 AM
P5	#42	Aliens are attacking!			OPEN	0 requests	11/Jul/20 8:07 AM
P1	#41	All your base are belong to us	Refunds Service	Billing Team	OPEN	1 request	09/Jul/20 4:40 AM

- Read [Working with major incidents](#) for more details.

Getting Opsgenie API information

When integrating Jira Service Management or Jira Software cloud with Opsgenie, you'll need to provide the Opsgenie API URL and key.

Opsgenie API URL

Opsgenie can have two different API URLs, and the one you need depends on where your Opsgenie instance is located.

US

- <https://api.opsgenie.com>

EU

- <https://api.eu.opsgenie.com>

Opsgenie API key

To generate an Opsgenie API key:

1. In Opsgenie, go to **Settings > API key management**.
2. Select **Add new API key**.
3. Enter a name for the API key and select the following access rights for this API key:
 - Create and Update Access
 - Read Access
 - Delete Access
 - Configuration Access

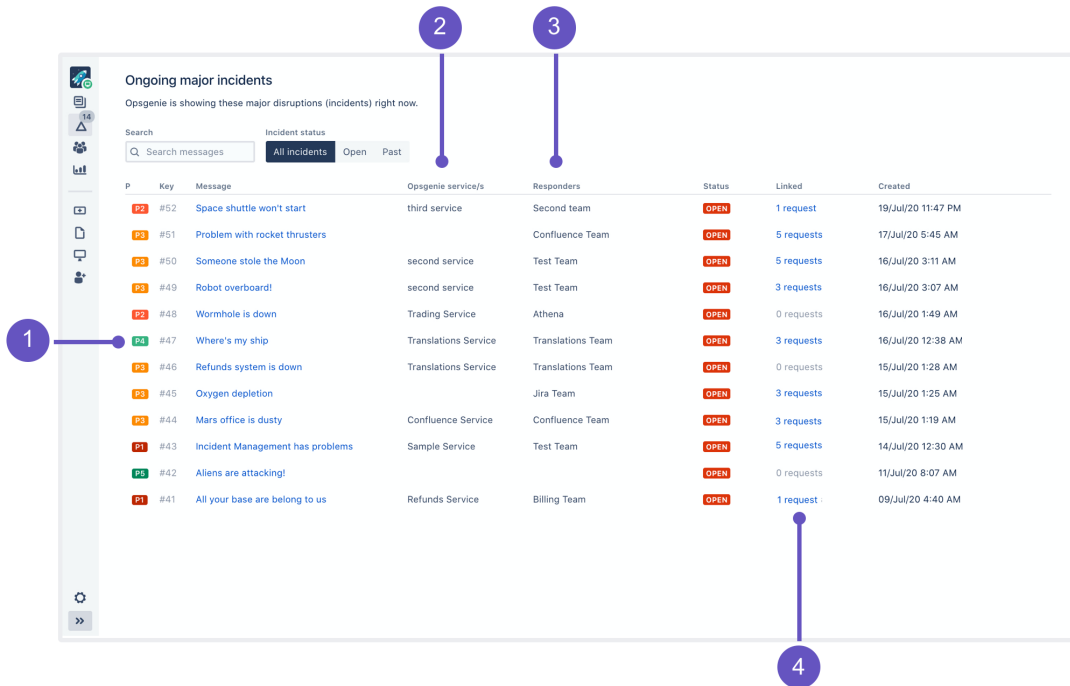
For more info, see [API key management](#).

Working with major incidents

Once you set up incident management, you'll get access to a new page Major incidents. It shows ongoing major incidents from Opsgenie, and allows your agents to view them, stay up to date, and link requests to these incidents, just to give the Opsgenie teams more context.

Viewing major incidents

To view major incidents, select **Major incidents** in the left navigation bar.



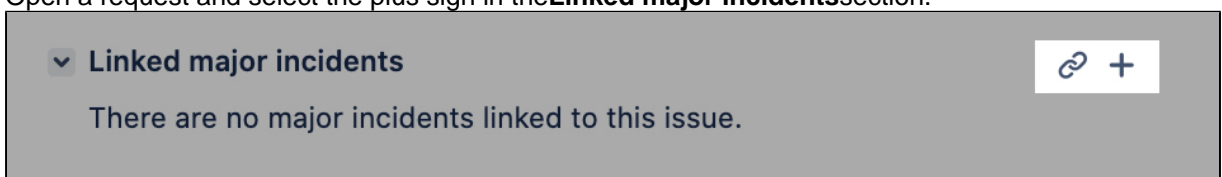
1. **Priority, key, and message:** Details about your incident. You can click the incident message to view it in Opsgenie in more detail.
2. **Affected Opsgenie service:** Services are configured on the Opsgenie side and allow you to properly categorize an incident.
3. **Responders:** That's an Opsgenie team that will be responsible for resolving your incident.
4. **Linked requests:** Requests that are linked to an incident. You can click this link to open a view where you can comment on all linked requests, reassign them, or complete some other actions.

Creating major incidents

You should create a major incident when you keep getting requests about some service being down, and need someone to take action on resolving it. The incident will be created in Opsgenie (and shown in Jira Service Management), and will alert the right Opsgenie teams right away.

To create a major incident:

1. Open a request and select the plus sign in the **Linked major incidents** section.



2. Fill in the details of your incident. We'll show you ongoing incidents, just in case someone already created a similar incident.

- When you're ready, click **Create**. The major incident will be linked to your request, and will also appear on the **Major incidents** page.



Linking requests to major incidents

When you keep getting requests about the same incident, you can link all of them to a major incident. This will help you identify the severity of the problem, and will give the Opsgenie teams more context about an issue. It's also useful for other agents.

Linking a single request

To link a request to a major incident:

- Open a request and select the Link sign in the **Linked major incidents** section.
- Search for a major incident or select it from the list.

Linking multiple requests

You can also link multiple requests to a single major incident at once by using agent queues.

To link multiple requests to a major incident:

- Open your agent queue.
- Select your requests.
- Select **Link to major incident**, and choose the incident.

Linking other related requests

It often happens that a request is linked to other requests (*is blocked by, is duplicate, etc.*). When you're linking such a request to a major incident, you can choose whether other linked requests should be linked to this incident as well.

- Open a request that is linked to some other request.
- Link it to a major incident.
- You'll be asked whether you want to link other requests to this incident. Select them, or click **Cancel**.
- Likewise, when you unlink a request from an incident, we'll ask you what to do with the related requests.

Interacting with linked requests

When multiple requests are linked to a major incident, you can easily leave a comment, reassign them, or complete other actions on all of these requests at once, instead of opening them one by one.

- Open the **Major incidents** page.
- Next to an incident, click the hyperlink in the **Linked** column.
- Select requests, and choose an action you want to complete.

Requests linked to #53



<input checked="" type="checkbox"/>	8 selected	Link to major incident	Link	Comment	Assign	Watch	Delete
<input checked="" type="checkbox"/>	CHARLIE-10	There's an old car in the lower orbit	Friendly Robot	Captain Joe	WAITING FOR SUPPO...		21/Feb/20
<input checked="" type="checkbox"/>	CHARLIE-8	When can we expect travels to Saturn?	Captain Joe	Unassigned	WAITING FOR SUPPO...		30/Oct/19
<input checked="" type="checkbox"/>	CHARLIE-7	Mayday, I'm lost in space	Master Researcher	Unassigned	WAITING FOR SUPPO...		30/Oct/19
<input checked="" type="checkbox"/>	CHARLIE-6	Red dust gets everywhere	Master Engineer	Unassigned	WAITING FOR SUPPO...		30/Oct/19
<input checked="" type="checkbox"/>	CHARLIE-5	Cleaning service in the Mars base	Droid	Unassigned	IN PROGRESS		30/Oct/19
<input checked="" type="checkbox"/>	CHARLIE-4	Requesting more flights to Mars	Crazy Scientist	Unassigned	IN PROGRESS		30/Oct/19
<input checked="" type="checkbox"/>	CHARLIE-3	Engine makes weird noises	Friendly Robot	Unassigned	PENDING		30/Oct/19
<input checked="" type="checkbox"/>	CHARLIE-1	What am I looking at?	Lieutenant Woolf	Unassigned	WAITING FOR SUPPO...		30/Oct/19

Filtering out requests

There are also a few ways that let you find unlinked requests (to check if they need an incident) or requests linked to a specific incident. This can come in handy if you have many requests or incident and cant easily find them on the Major incidents page.

Finding unlinked requests

1. Open agent queues.
2. In the top-right, change the filter to **Unlinked requests**.

Finding requests linked to a specific incident

1. Open an incident in Opsgenie, and copy its ID from the URL. It will look something like this:

```
582f7731-d00f-4827-9a2d-19e380b26205
```

2. In Jira, open the advanced search and use the following JQL query:

```
"Linked major incidents" in ("582f7731-d00f-4827-9a2d-19e380b26205")
```

Creating a queue with linked or unlinked incidents

1. Go to **Queues > New queue**.
2. Switch to **Advanced**.
3. Use one of the following JQL queries to either find linked or unlinked incidents:

```
"Linked major incidents" is not EMPTY
```

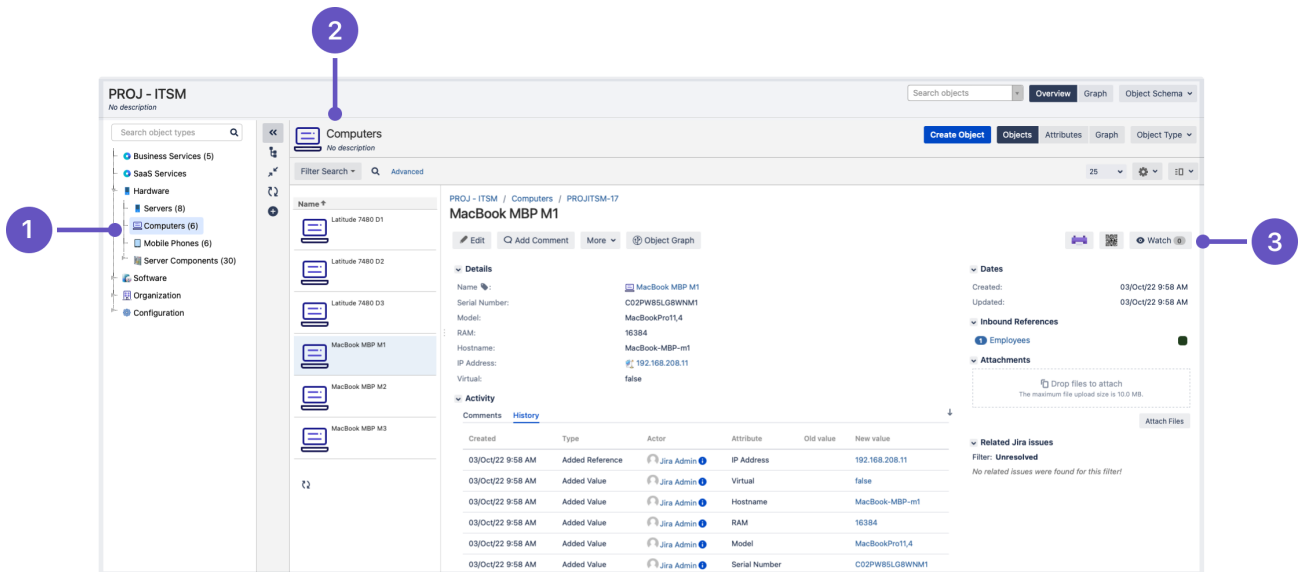
```
"Linked major incidents" is EMPTY
```

More resources

- [Opsgenie documentation](#)

Managing your assets with Assets

Assets lets you manage your assets and configuration items in Jira Service Management, and interact with them through requests. It's a database of objects—digital representations of your assets, ranging from hardware and software to employees or various configuration items. You can make your objects whatever you need them to be, and have them displayed in and be affected by issues in Jira.



1. **Object types:** List of object types grouping your assets, organized into a clear tree structure.
2. **Objects:** Your actual assets or configurations items, with all details added through attributes. From simple names to type-specific items, like serial numbers, models, and pretty much anything you need.
3. **Object actions:** You can keep your physical assets in check by printing labels and QR codes, view their dependencies with other objects, and see which Jira issues they're on.

Get started with Assets

Here's list of pages to get you started:

- [Getting started with Assets](#)
- [Administering Assets](#)
- [Working with object schemas](#)
- [Working with object types](#)
- [Working with objects](#)
- [Importing your data](#)
- [Searching for objects](#)
- [Working with reports](#)

Advanced usage guides

If you're looking for more advanced knowledge, including Assets app development, REST APIs and Java docs, see [Advanced usage guides](#).

Getting started with Assets

This page will explain the basics of Assets, and it's then up to you how much deeper you'd like to dive in.

What is Assets?

Assets is used for asset and configuration management. Its like a database that stores everything you own from hardware and software to offices, keyboards and licenses, Assets can store whatever you need it to store. Its really up to you, though were guessing the most popular items will be something like:

- **Business services** (billing, email, payroll)
- **Hardware and software** (servers, computers, mobile phones, software licenses)
- **Employees** (John, Mary, Mathias)
- **Offices** (Sydney, Melbourne, Stockholm)
- **Server components**(RAM, network adapters, hard drives)

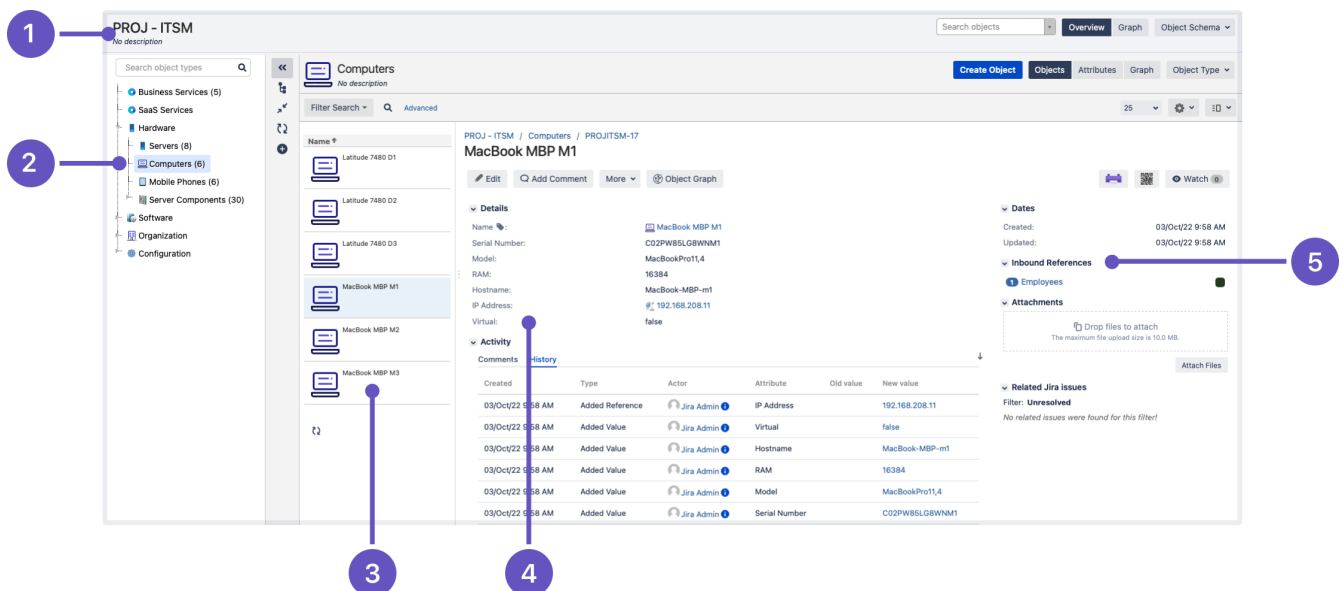
But isnt Jira all about issues? Why assets?

Assets is more than just a Jira app. To be frank, under the hood, it's a perfectly complete and powerful product that brings new features and possibilities. But it looks like an app, and we've bundled it into Jira Service Management, so let's treat it as an app for now. As for the issues imagine you create a Jira issue for your laptop you give it a name, maybe some description, and fill in all these custom fields about brand, model, serial number, RAM, storage, and boom you have an asset. Do you see the similarity?

So are my assets issues?

They look similar, but theyre not your typical issues. Assets has its own structure for data that you put into it. Understanding this structure will help you in the future, but if you want a quick start, its really about object schemas, object types, and objects, together with their attributes and relations between them.

Heres what it looks like in Assets:



1. **Object schema:**It works like a map that holds all of it together. In the Jira world, that would be a project.
2. **Object types:**Containers for your objects (assets), each with its own characteristics, like specific issue types (Computers, Mobile phones).
3. **Objects:**Your actual assets, whatever they are. Similar to Jira issues or requests. Put them into object types, and get going (MacBook MBP M1, Latitude 7480).
4. **Attributes:**What defines your object types (objects inherit them). Like fields or custom fields (Name, RAM, Storage).
5. **References:**Relations and dependencies between objects. If you own a laptop, you have a reference to it (you own it). Like issue links, though they serve a different purpose.

The best way to start quickly is to use the [starting from scratch tutorial](#). It will show you the basics, essentially taking you from zero to hero. If you don't have time and want to see a ready object schema, take the [ITSM tutorial](#).

License

Assets is part of Jira Service Management Data Center 4.15 and later, but you can also download it at no extra cost for earlier versions of Jira Service Management Data Center, and also for Jira Software Data Center. If you're on one of these earlier versions, it's best to [see our FAQ](#) for answers to any questions you might have.

Learn more about Assets

If you're curious to learn more, here are some pages for you:

- [What is Assets?](#) an overview of the most important concepts and features: object schemas, object types, and objects.
- [Assets onboarding guide](#) if you'd rather get a one-pager with everything that matters. We'll guide you through the basic concepts, tell you how to start, and give some tips and tricks on what data to include and how to keep it accurate in the future. That's exactly what you need if you've never used Assets before.

To get some practice, see these tutorials:

- [Tutorial: Starting from scratch](#)
- [Tutorial: How Jira issues affect Assets objects \(ITSM\)](#)

Start working with Assets

If you'd rather get working with Assets right away, here are the pages you'll need:

- [Administering Assets](#)
- [Working with object schemas](#)
- [Working with object types](#)
- [Working with objects](#)
- [Searching for objects](#)
- [Working with reports](#)
- [Importing your data](#)

Advanced usage guides

If you're looking for advanced knowledge, including Assets app development, REST APIs and Java docs, see [Advanced usage guides](#).

What is Assets?

You can jump right into Assets, but understanding what it's made of and how it's treating your data will make your future work easier. Here you'll find the most important concepts and information about useful features.

Jump to...

- [Understanding your data in Assets](#)
- [Interacting with objects through requests](#)
- [Importing your data](#)
- [Reporting](#)
- [Searching and IQL](#)

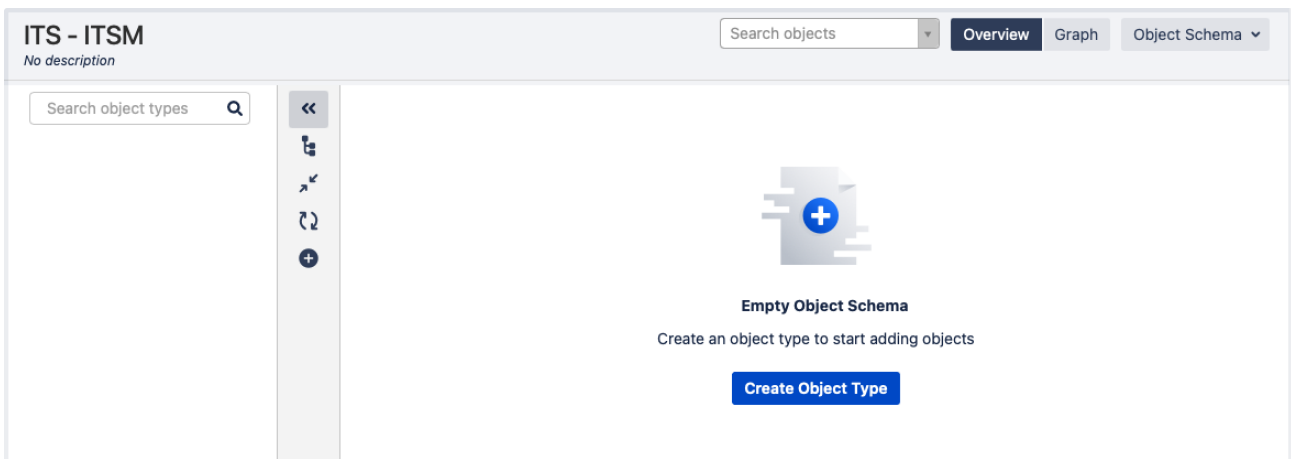
Understanding your data in Assets

To understand how your data is structured in Assets, you'll need to learn about its main building blocks.

Object schemas

An object schema is a collection of object types (groups of objects) and their attributes (fields), as well as underlying objects and references between them. It also includes your configuration items, such as roles, icons, statuses, and automation rules. Object schemas work like maps that hold all of it together, and are the first thing you'll create in Assets. In the Jira world, object schemas would be your projects, but you're not limited to one schema per project.

An empty object schema, without any object types or objects. A fresh start.

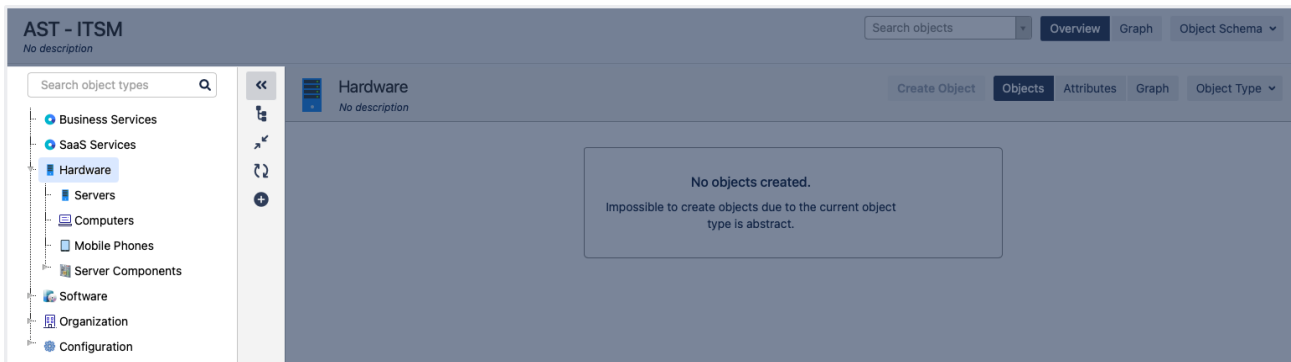


Learn more at [Working with object schemas](#).

Object types

Once you have your schema ready, you'll create a set of object types that will group objects your actual assets. Rather than a single PC, your object types would be Computers, Hardware, Software, Employees, and so on. Object types can be whatever you want them to be as Assets is very open and flexible. In the Jira world, object types would be your issue types configuration (characteristics of bugs, incidents, problems).

An object schema with object types, but without objects yet

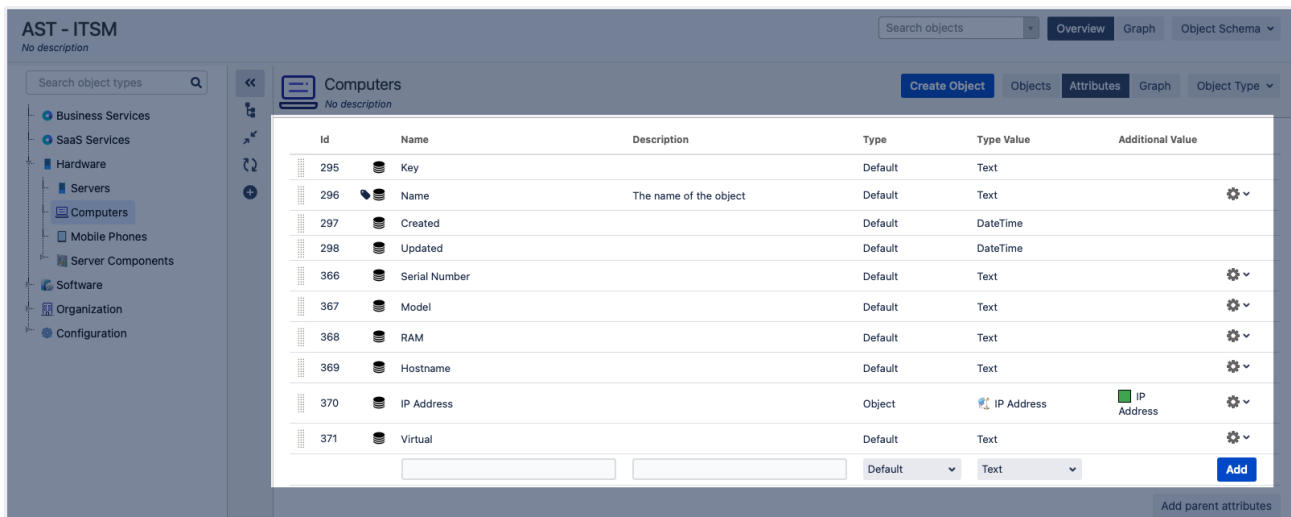


Learn more at [Working with object types](#).

Attributes

Attributes is what defines your object types (and underlying objects that use these attributes). They work like fields to which you add data. You'd choose attributes for your object types, and then the same attributes need to be filled in for underlying objects, whenever someone creates them. They come in different shapes and sizes: text, dates, references to other objects, and so on. In the Jira world, attributes would be issue fields or custom fields.

Attributes defined for an object type Computers

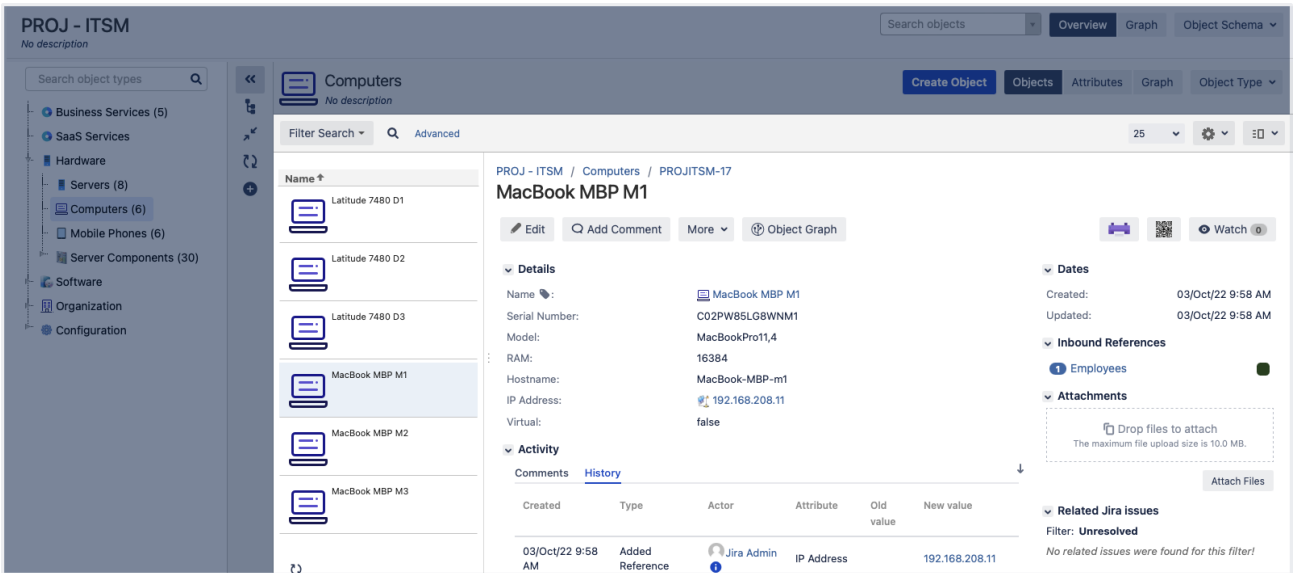


Learn more at [Adding attributes](#).

Objects

Objects are your actual assets. Every object is a digital representation of anything that you're mapping in Assets, be it a specific computer, employee, office they work at, or even a license for your software. You can create as many objects as you like, and group them within object types that represent their characteristics. In the Jira world, objects would be your issues or requests, and just like you wouldn't mix epics with bugs, you shouldn't put employees into hardware.

Object schema with objects and their details defined through attributes

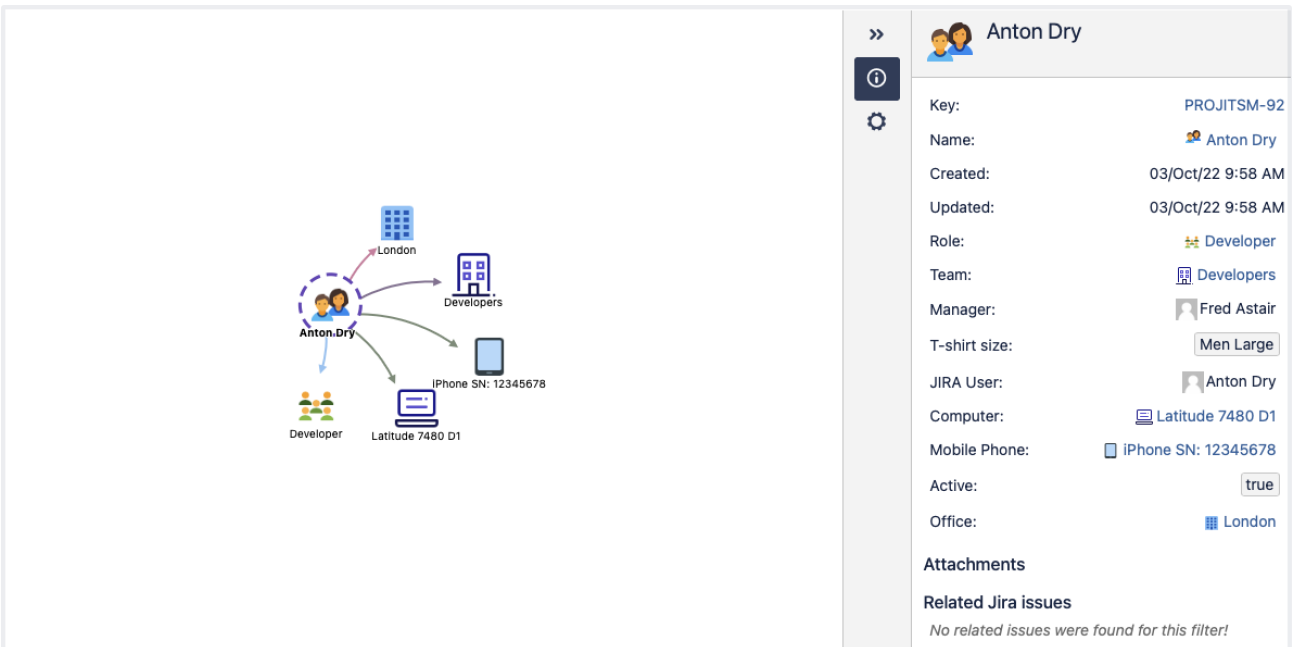


Learn more at [Working with objects](#).

References

References are types of attributes that let you create references to other objects. Information about a specific object can be useful, but viewing its relations and dependencies is even better. If you look at references of an employee, you know right away the office they work at, hardware they're using, role they belong to, or mobile phone they've been assigned. You can view them in the graphical viewer that lets you see all of it at a glance. In the Jira world, references would be issue links (blockers, duplicates), but they serve a bit different purpose in Assets.

An employee with references to other object types and objects

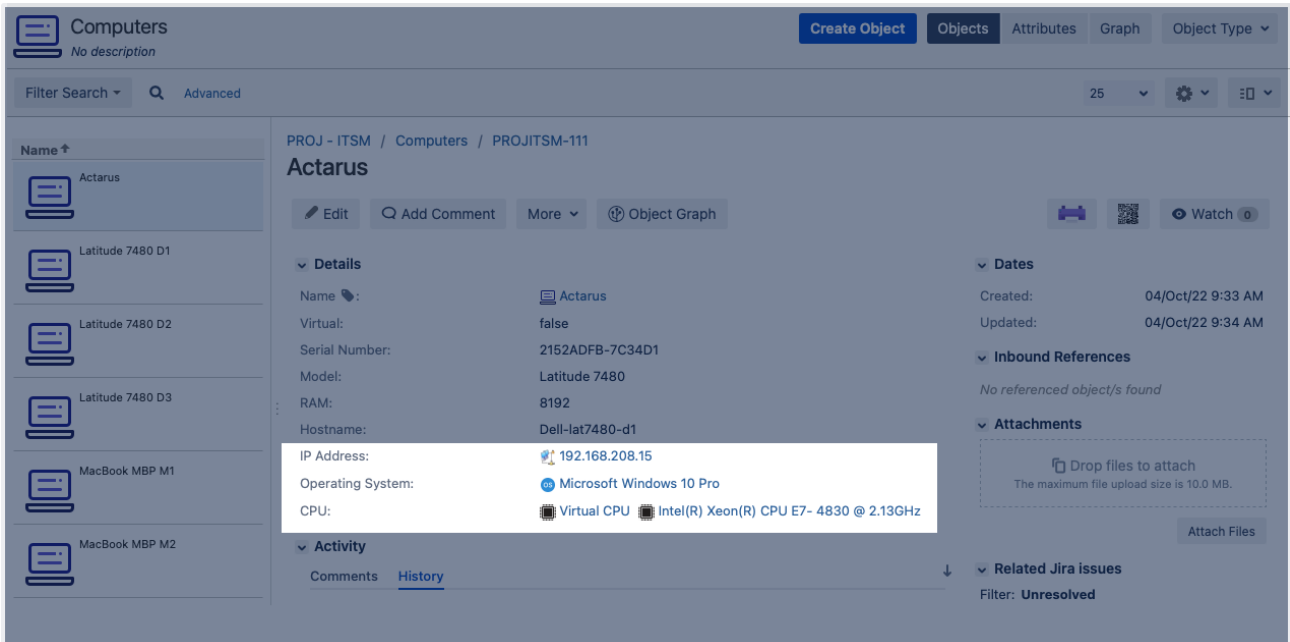


Learn more at [Viewing references](#).

While we're at references, you can have two types:

Outbound references (outgoing)

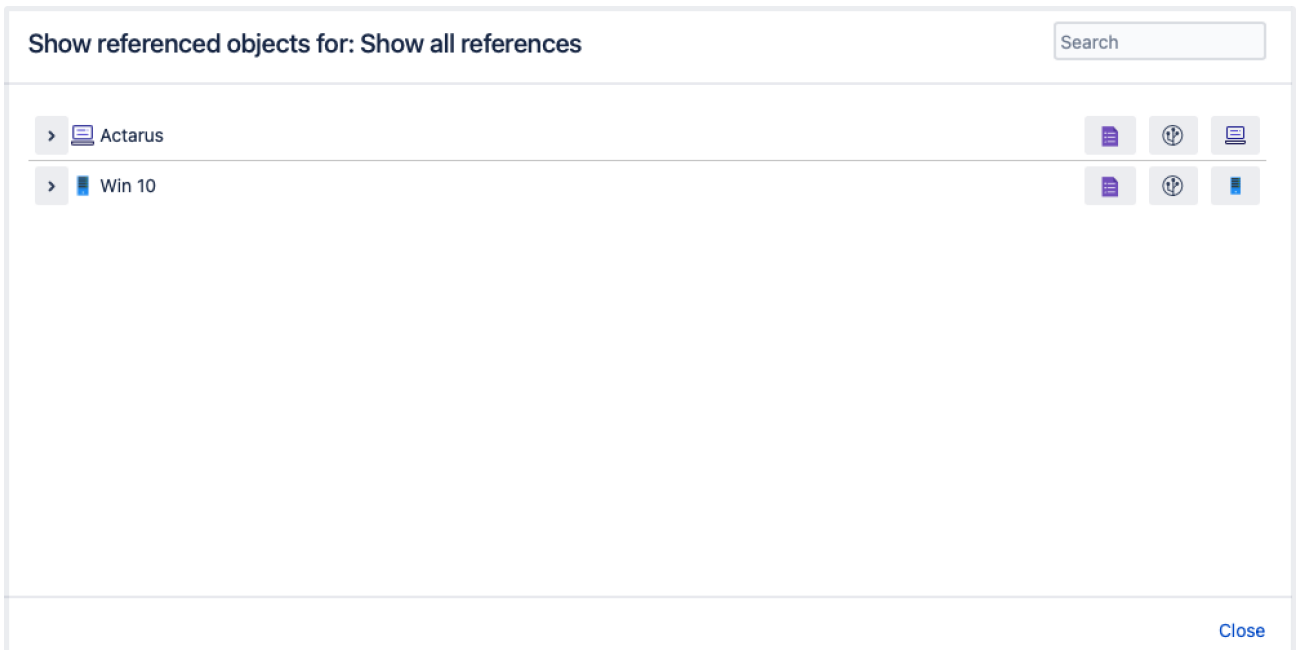
Outbound references are attributes within an object that point to another object. For example, a Host object might have references to other important parts of its system, such as operating system, CPU, network interface, or scanning information.



Inbound references (incoming)

Inbound references are references that come from other objects. Each outbound reference creates an inbound references on the object it's referencing, which lets you see how different objects are connected across your object schema.

For example, in the image above, you had the host Actarus referencing a number of CPUs, including one called *Intel(R) Core (TM) i7-4700HQ CPU*. If you clicked on this CPU reference, you'd see the object itself and a list of its inbound references, including the host Actarus.



Interacting with objects through requests

Having a database of assets is awesome, but there's a reason we've added Assets to Jira Service Management.

Showing object details on issues

You can add your objects to Jira issues through Assets custom fields. This could be adding the affected business service to your incident issues or adding a computer to a hardware request. Whatever you do, extra information about affected assets will give everyone involved more context.

Details

Type: 🚫 Problem Status: OPEN (View Workflow)

Labels: None Resolution: Unresolved

Affected Server:

exchange.prod.app1

Status: RUNNING

Environment: Production

Server Owner: Fred Astair

Affected Business Service:

Email Service

Status: RUNNING

Business Service Owner: Wes Shults

Service Group Phone: +46 3284 234 442

Importance: Critical

Learn more at [Adding Assets custom fields](#).

Affecting your objects

Have you noticed the RUNNING status on the screenshot above? Well, things break. Assets lets you add conditions, validators, and post functions to your workflow transitions so that your issues affect related objects when they transition through the workflow. And what about broken parts that can also be your objects? Decommissioned, for good.

Details

Type: 🚫 Problem Status: WAITING FOR CHA... (View Workflow)

Priority: ↓ Low Resolution: Unresolved

Labels: None

Affected Server:

exchange.prod.app1

Status: RUNNING

Environment: Production

Server Owner: Fred Astair

Affected Server Components:

Apc Network Management Card 2

ID: APC-2-2

Server: exchange.prod.app1

Affected Business Service:

Email Service

Status: RUNNING

Business Service Owner: Wes Shults

Service Group Phone: +46 3284 234 442

Importance: Critical

Details

Type: 🚫 Problem Status: COMPLETED (View Workflow)

Priority: ↓ Low Resolution: Unresolved

Labels: None

Affected Server:

exchange.prod.app1

Status: RUNNING

Environment: Production

Server Owner: Fred Astair

Affected Server Components:

Apc Network Management Card 2

ID: APC-2-2

Status: DECOMMISSIONED

Affected Business Service:

Email Service

Status: RUNNING

Business Service Owner: Wes Shults

Service Group Phone: +46 3284 234 442

Importance: Critical

Learn more at [Adding Assets functions to workflows](#).

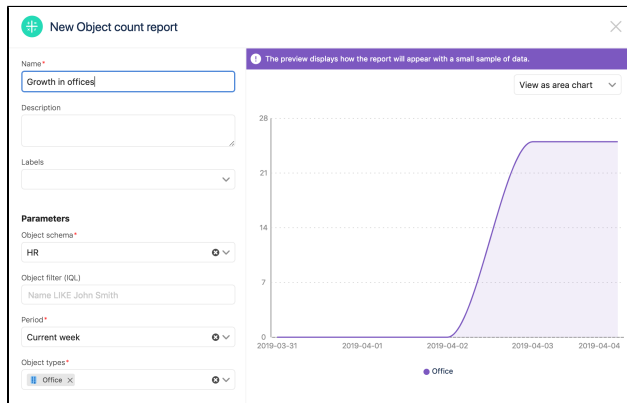
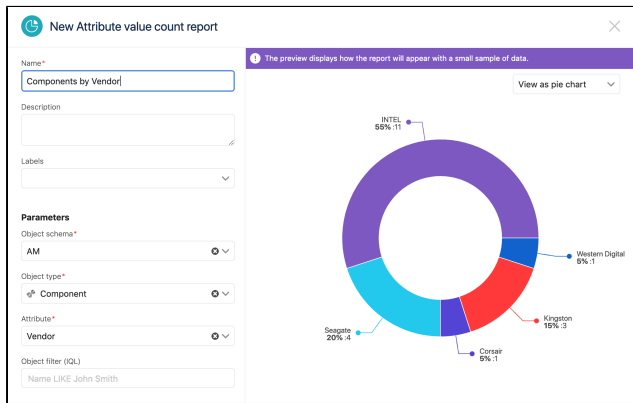
Importing your data

You're free to handle your objects however you like, but entering everything manually might not be the best idea if you have a lot of assets. To help you get going, we have several tools that will either scan your environment automatically or facilitate your manual imports.

Learn more at [Importing your data](#).

Reporting

You can also use dynamic reports to view information about your object schema in a variety of formats.



Learn more at [Working with reports](#).

Searching and IQL

Searching for and filtering objects in Assets is similar to Jira: you can use basic search (free text) or something that's called Assets Query Language. Although different in use, the idea behind it is similar to the Jira Query Language (JQL). By using IQL and placeholders, you can not only search for objects, but also create advanced queries and decide what objects should be included where.

Learn more at [Searching for objects](#).

Tutorial: Starting from scratch

This tutorial will guide you through the most important aspects of Assets, essentially taking you from zero to hero. Here's what you'll learn:

- Create a basic object schema
- Create object types
- Add attributes to object types
- Create objects for each object type
- Link your object schema to a Jira project
- Add your objects to Jira issues

Before you begin

The tutorial takes you through all the basic actions you'll need to know about in Assets. This, however, has some disadvantages:

- You'll be adding custom fields to screens in Jira, which requires admin permissions.
- You'll be creating object types, attributes, and objects manually, which might take some time.

If you don't have sufficient permissions (but can create projects) or want to see a schema pre-filled with data, you can also try the [ITSM tutorial](#) that uses sample data to create a nice object type schema.

In this tutorial

Here are all the steps you'll follow in this tutorial:

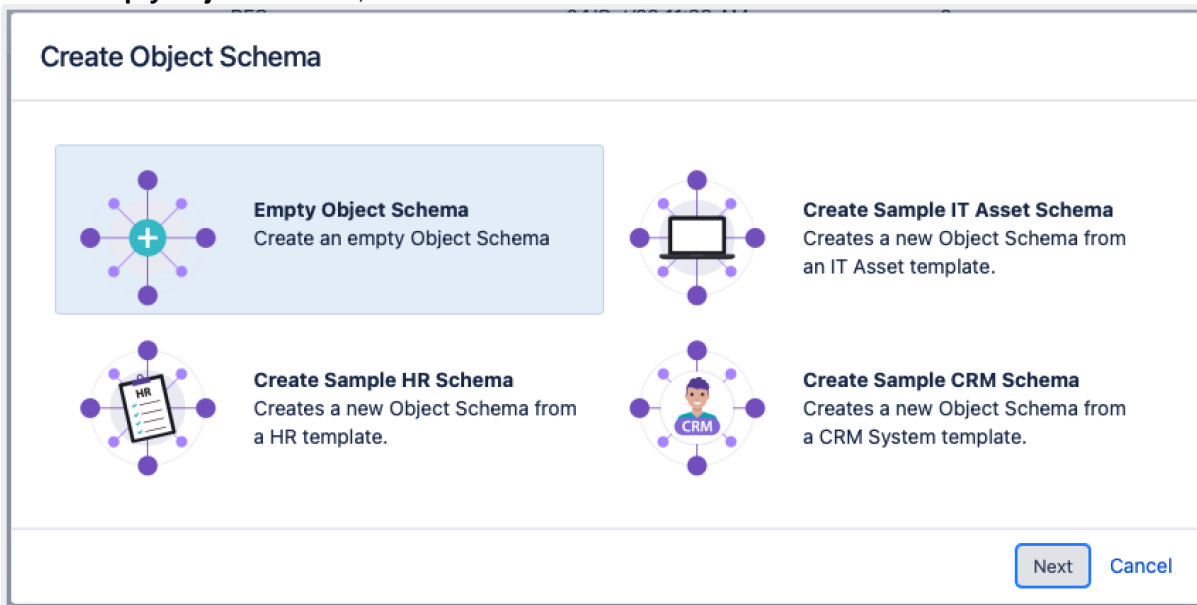
- [1. Build your first object schema](#)
- [2. Create object types](#)
- [3. Add attributes](#)
- [4. Create objects](#)
- [5. View object graphs](#)
- [6. Link your object schema to a Jira project](#)
- [7. Add object to your requests](#)

1. Build your first object schema

An object schema is a collection of object types and their attributes, objects, and references between them. It also includes your configuration items, such as roles, icons, statuses, and automation rules. Object schemas work like maps that hold all of it together.

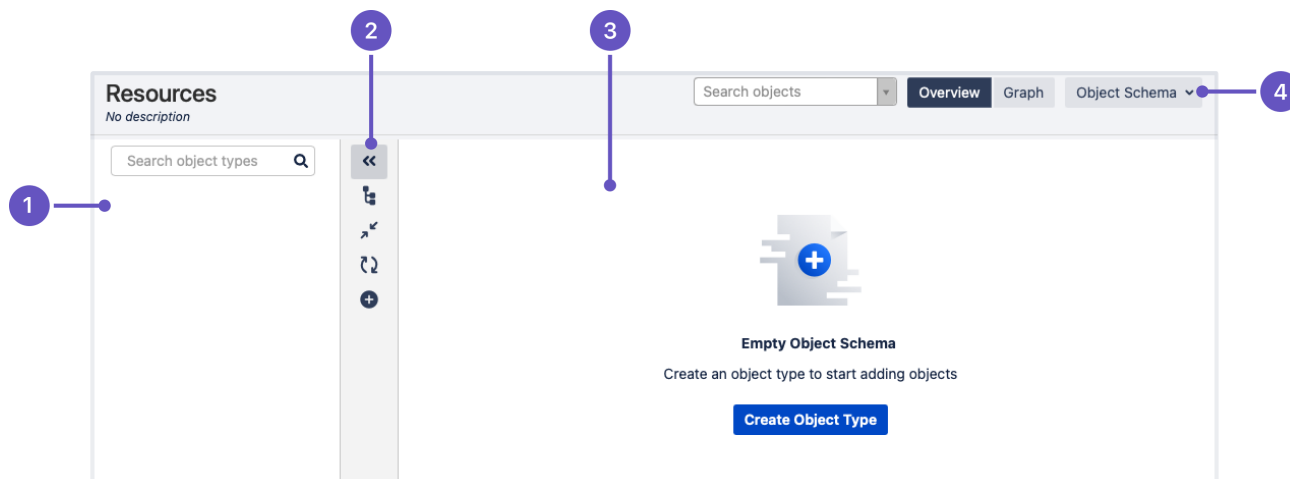
To create a new object schema:

1. In the Jira top navigation bar, select **Assets > Assets object schemas**.
2. Select **Create an object schema**.
3. Select **Empty object schema**, as shown below.



4. Give it a name, say "Resources".
5. Select **Create** and voil! You can now access your object schema under **Assets > Object schemas**.

Here's some explanation for what you should be seeing on your screen:



1. **Object types:** When you start creating object types, they'll appear here in a tree structure.
2. **Vertical menu:** Some actions for your schema. You can use it to expand or collapse the object type view, or create new object types.
3. **Create:** Here you'll create your first object type.
4. **Object schema actions:** Some configuration and actions for your object schema.

Next up

When you're ready, go to [2. Create object types](#).

2. Create object types

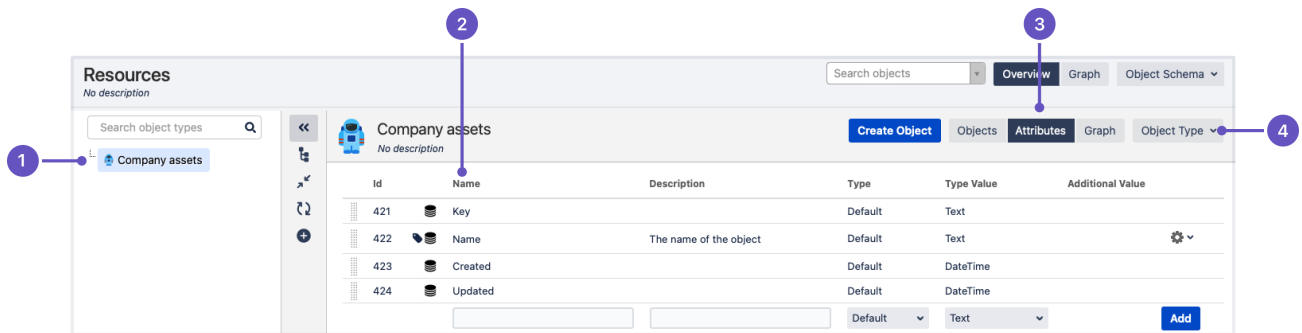
An object type groups objects that use the same kind of information, conveyed through their common attributes. Rather than a single PC, your object types would be Computers, Hardware, Software, Employees, and so on.

Create your first object type

To create the first object type:

1. In your object schema, select **Create**.
2. Call your object type *Company assets*. You can select an icon for the type and enter a description.
3. Set its parent to **None**. We'll make it an individual, stand-alone object type.

Here's what you should see:



1. **Object type:** Your first object type, congratulations!
2. **Attributes:** Default attributes (fields) created for your object type. We'll add some more in the next steps.
3. **Object type view:** Here you can switch between viewing objects, attributes, and a graph for your selected object type.
4. **Object type actions:** Here you can configure your object type, or create a new one.

To create another object type:

1. Select the **Object type** dropdown at the top-right, and select **Create**.
2. Call your object type *Server Type*.
3. Set its parent to *Company Assets*. It will be a child object type.
4. Select Object Type at the top-right, and again select **Create**.

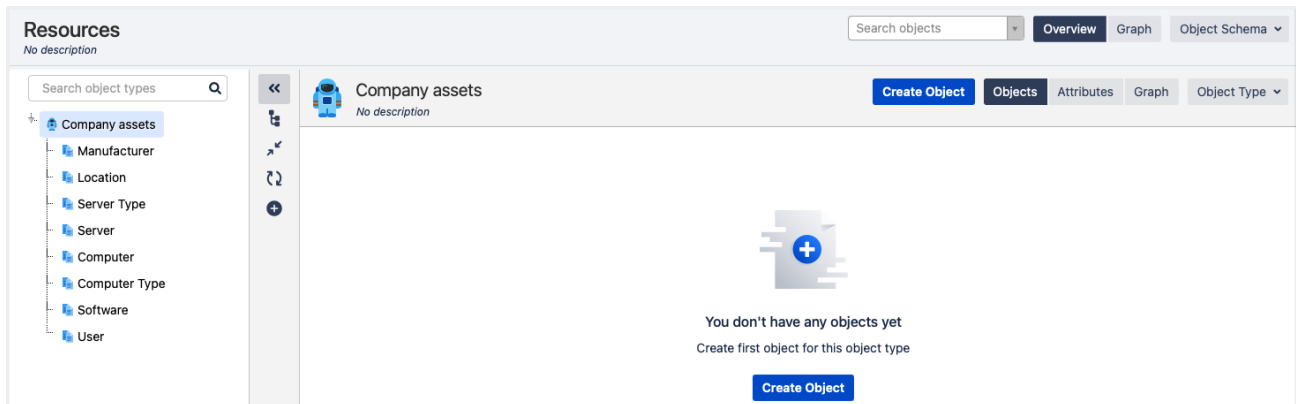
Create some more

Create more object types, like shown in the image below. Set the parent for all these object types to *Company assets*.

Parent object type	Child object types
Company assets	Manufacturer
	Location
	Server Type
	Server
	Computer
	Computer Type
	Software



Here's what your schema should look like after creating these object types. Hopefully you've picked better icons than we did:



Next up

When you're ready, go to [3. Add attributes.](#)

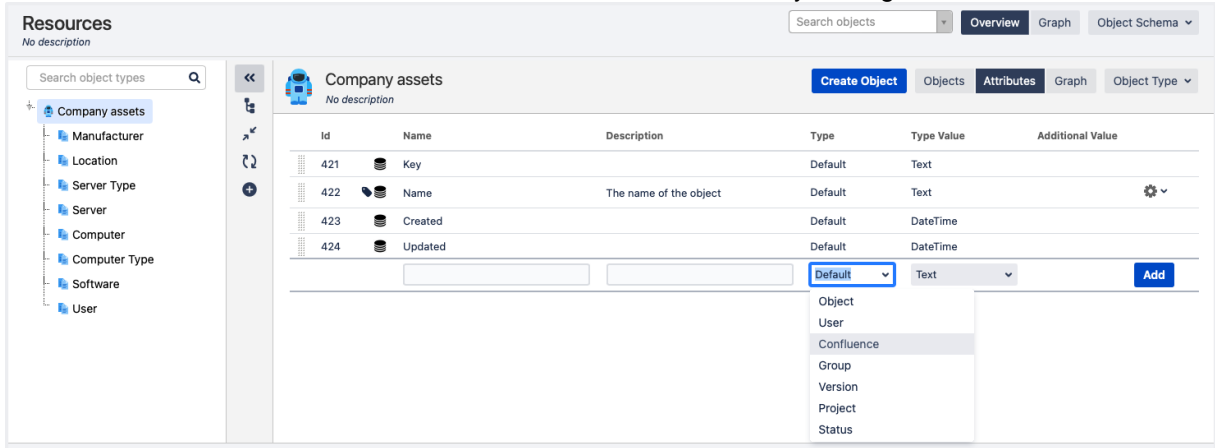
3. Add attributes

Attributes are details that define your object types. You can view all attributes of an object type by selecting **Attributes** at the top-right. Assets creates a few default attributes when you create an object type.

View attributes for an object type

To view attributes:

1. When viewing an object type, switch to **Attributes** at the top-right.
2. You'll see the default attributes, and will be able to add new ones by adding them to the list.



Each attribute needs the following to be defined:

Attribute value	Description
Name	Name of your choice.
Type	Type of an attribute. Each type does a different thing, but we'll get to that with examples. Just to let you in on some secret, type <i>Object</i> would be responsible for linking to other objects.
Type value	Value for a type. The options here would depend on the type you've chosen. For example: <ul style="list-style-type: none"> • Default: includes Text, textarea, Date, Email, etc. • Object: Other objects or object types (in this tutorial, this would include Location, User, Software, or any other of the object types you already created)
Additional value	This is some additional information for the attributes, and it also depends on the types you've chosen. For example, if you're configuring an attribute of type <i>Object</i> , in this field you'd specify a specific reference.

Add attributes to your object types

Now, to get a little practice, create attributes for all your object types as shown in the table below.

Object type you have	Attribute name	Type	Type value	Additional value
----------------------	----------------	------	------------	------------------

User	Email Address	User	jira-users/any other group	-
	Phone Number	Default Object	Text Location	- Reference
	Location			
Software	Manufacturer	Object	Manufacturer	Reference
Manufacturer	Street Address	Default	Textarea	-
	Phone Number	Default	Text	-
Server	Warranty Expires	Default	Date	-
	Manufacturer	Object	Manufacturer	Reference
	System Owner	Object	User	Belongs
	Type	Object	Server Type	Type
	Location	Object	Location	Deployed
	Installed Software*	Object	Software	Installed
Computer	Warranty Expires	Default	Date	-
	Manufacturer	Object	Manufacturer	Reference
	RAM	Default	Text	-
	Type	Object	Computer Type	Type (Enter "Type" to create a new reference type)
	Display Size	Default	Text	-
	User	Object	User	Uses

* Additionally, for the Server > Installed Software attribute, select the cog wheel at the far right and select **Configure**. Under the **Cardinality** tab, select the *Unlimited* checkbox. This allows you to specify more than one software installed on a server.

Next up

When you're ready, go to [4. Create objects](#).

4. Create objects

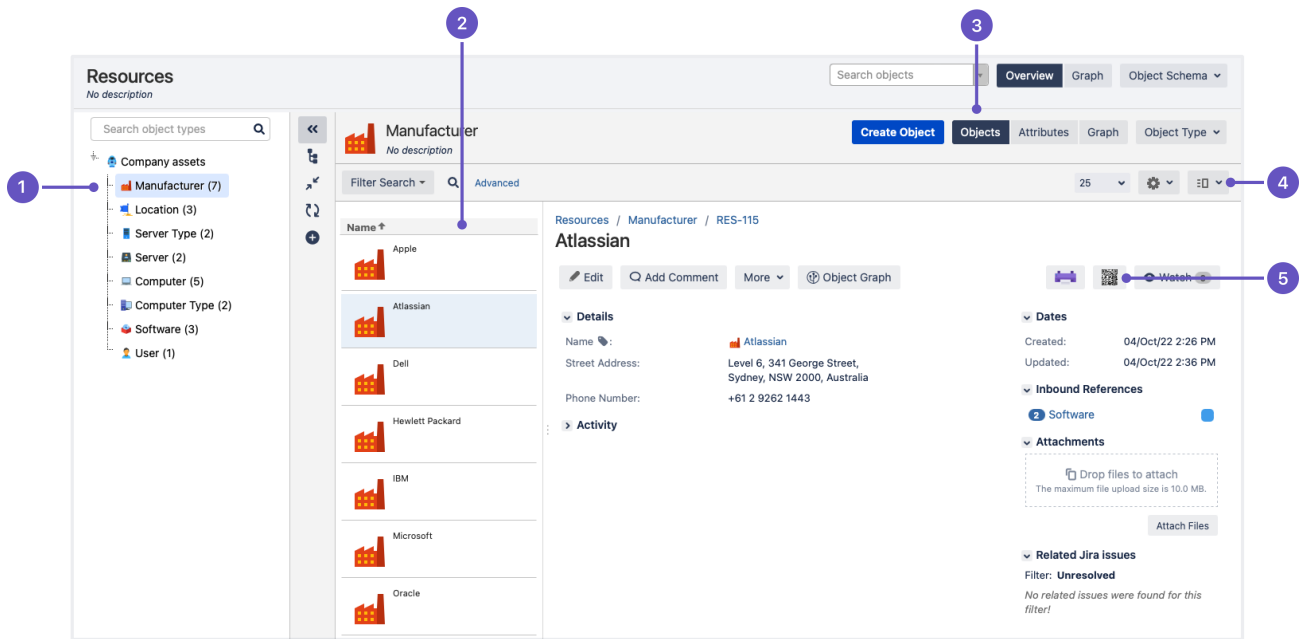
Objects are your actual assets. Every object is a digital representation of anything that you're mapping in Assets, be it a specific computer, employee, office they work at, or even a license for your software.

To create an object:

1. Select an object type.
2. At the top-right, select **Create object**. You'll need to specify its name and attributes. The required attributes will depend on the object type you've chosen.
3. Create objects based on the table below, and try to come up with some sample attributes.

Object type	Objects
Manufacturer	Oracle Atlassian Apple Microsoft Dell IBM Hewlett Packard
Software	JAVA 1.8 Jira Confluence
Server Type	Virtual, Physical
Location	New York Stockholm Madrid
Computer Type	Laptop Desktop
User	[Your Name]
Computer	MacBook Pro (15") MacBook Air (13") ThinkPad (15") Latitude (15") Vostro (13")
Server	Dell PowerEdge VRTX HPE ProLiant DL580 Gen9

And some explanations once you're done creating objects:



1. **Object count:** Now your object types show the number of objects they contain.
2. **Objects:** Objects from the selected object type.
3. **View:** Switch between the different view types.
4. **Details:** Here, you can switch between the detail view and the list view.
5. **Print label and Print QR code** You can print a label or QR code and put it on your physical asset. When you scan the label or code, the asset details will open in Jira.

Next up

When you're ready, go to [5. View object graphs](#).

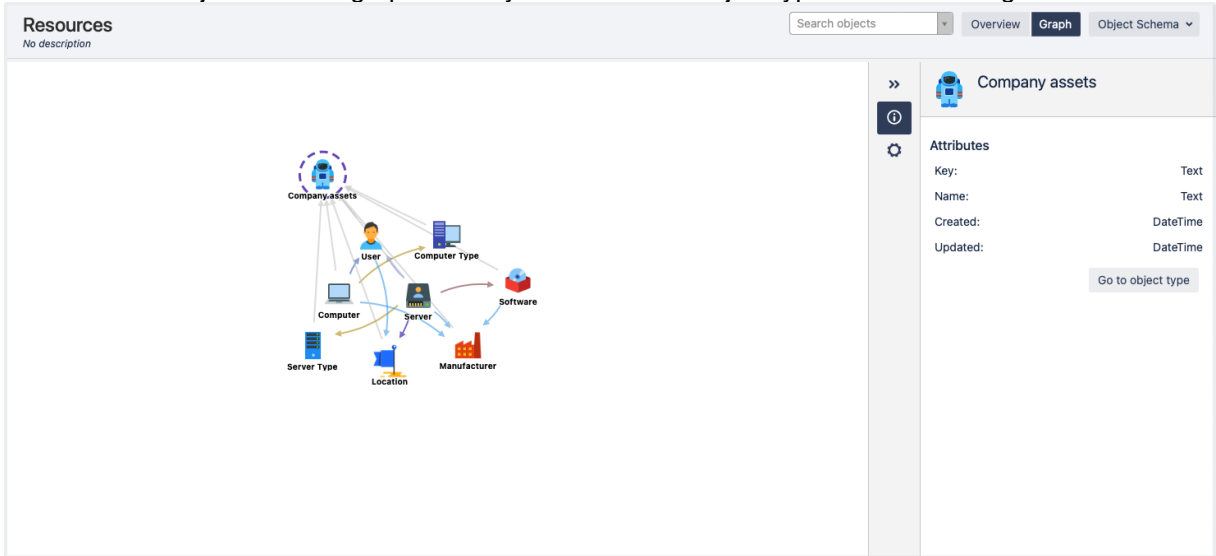
5. View object graphs

References are types of attributes that let you create references to other objects. Information about a specific object can be useful, but viewing its relations and dependencies is even better. You can do it by using a graphical viewer.

Object schema graph

To view a graph for your object schema:

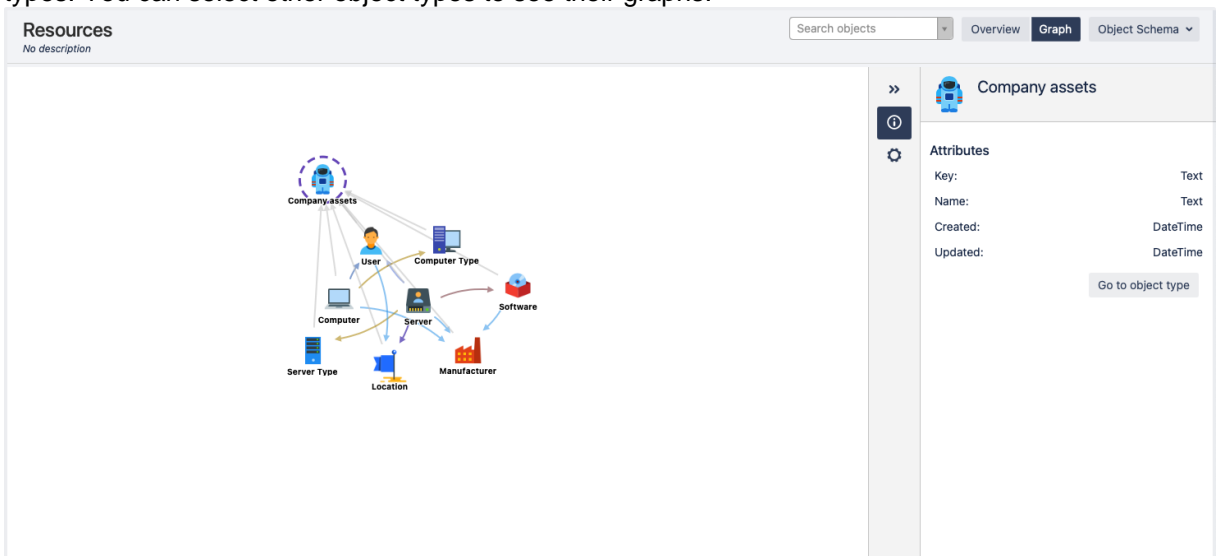
1. At the top-right, next to the object schema configuration, select **Graph**.
2. You'll see the object schema graph. It lets you see how all object types are knitted together.



Object type graph

To view a graph for your object type:

1. Select an object type, and select **Graph** next to the object type configuration.
2. You'll see the object type graph. It should help you see your object type's relations with other object types. You can select other object types to see their graphs.



Object graph

To view a graph for your object:

1. Select an object and select **Object graph**, just below its name in the detail view.

- You'll see the object graph that shows your object's relations with other objects and object types. Here you can additionally switch to the Tree view.

View Object Graph
Force **Tree**

The object graph shows a central 'Software' object connected to 'Confluence' and 'Jira'. 'Confluence' is further connected to two 'Server' objects and one 'Manufacturer' object. 'Jira' is connected to one 'Server' object and one 'Manufacturer' object. 'Atlassian' is also connected to 'Software'.

Referenced objects

Atlassian

Key: RES-115

Name: Atlassian

Created: 04/Oct/22 2:26 PM

Updated: 04/Oct/22 2:36 PM

Street Address: Level 6, 341 George Street, Sydney, NSW 2000, Australia

Phone Number: +61 2 9262 1443

Close

Next up

When you're ready, go to [6. Link your object schema to a Jira project.](#)

6. Link your object schema to a Jira project

You can link your object schema to Jira project by adding Assets custom fields to the right screens. These will let you select your objects when creating or editing issues.


Create a custom field

You'll include objects in your issues by adding a custom field to relevant screens. This custom field will then pull data about your objects (you'll limit the data to only specific objects or object types). In this example, we'll add all objects from the Server object type to the Create Issue screen.

1. In Jira, go to **Administration > Issues > Custom fields**.
2. Select **Add custom field**.
3. Select the **Assets object** field.
4. Name it "Affected Server", and add it to the Create issue screen for your project. You can add it to other screens, too.

Configure the custom field

Now, let's limit this custom field to show only a specific object type.

1. Select **More** () in the **Actions** column of your custom field (on the Custom fields page), and select **Configure**.
2. Select **Edit Assets configuration**.
3. Define the following:
 - **Object schema:** Mindville resources (or any other that you created)
 - **Filter scope (AQL):** `objecttype = Server`
 - **Type of interaction:** Select.
 - **Object attributes in issue view:** Select which attributes of Server you'd like to see in an issue when you create it.

Configure Assets Custom Field - Affected Server (customfield_10133)

Scope

Object Schema	Resources	i
Filter Scope (AQL)	objecttype = Server order by label ASC	i
Filter Issue Scope (AQL)	Not configured	i
Filter Assign Scope (AQL)	Not configured	i
Make Filter Assign Strict	<input type="checkbox"/>	i

User Interaction

Type of interaction	<input checked="" type="radio"/> Select <input type="radio"/> Checkbox / Radio button	i
Filter objects with attributes	Label x	i
Object attributes on Issue view	Name x User x Location x Installed Software x System Owner x	i
Multiple	<input type="checkbox"/>	i
Option to add all objects	<input type="checkbox"/>	i

Jira Service Management

Show object picker on customer portals	<input checked="" type="checkbox"/>	i
Force to show on Customer Issue details	<input type="checkbox"/>	i

[Return to Jira Custom Field](#)

Next up

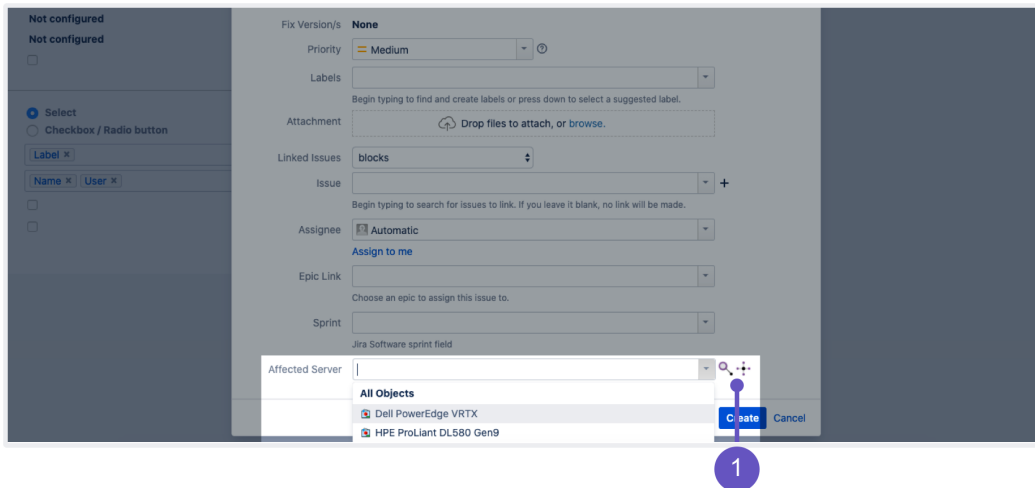
When you're ready, go to [7. Add objects to your requests.](#)

7. Add object to your requests

With Assets custom fields added to the right screens, you should be able to use them to select your objects.

Create a Jira issue

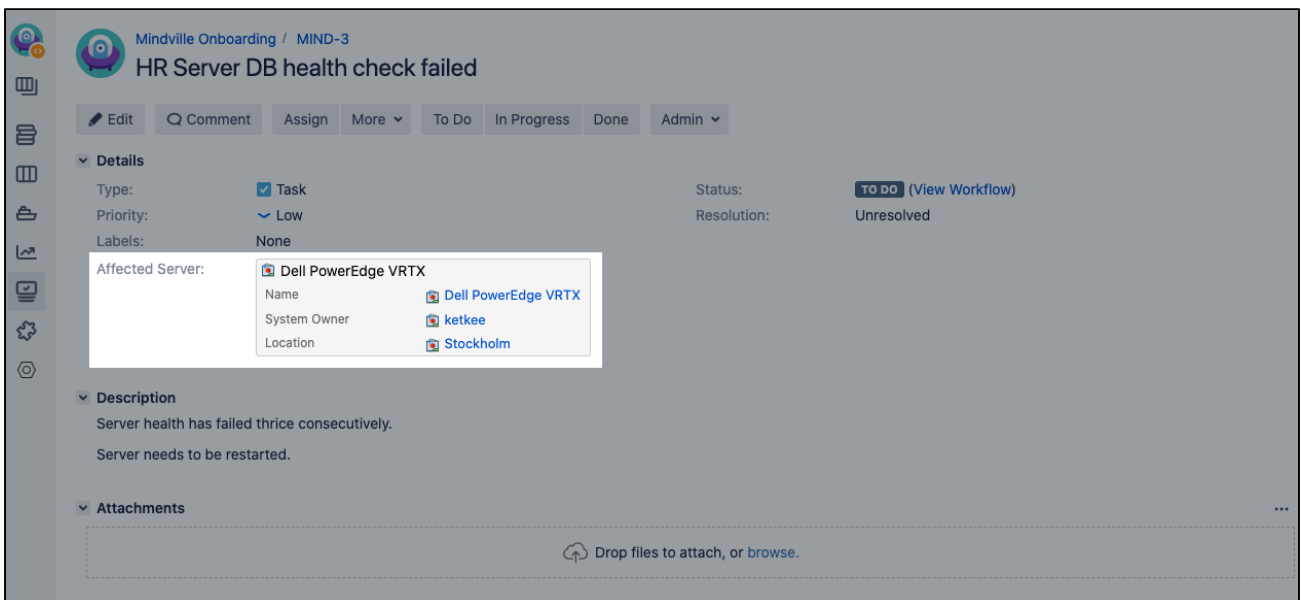
Create a Jira issue (make sure you've added your field to the Create issue screen for your project). You should be able to see your field like this:



1. **Object picker:** These little icons let you browse Assets objects or create new ones directly from your issue.

View object details on your issue

Once you created your issue, you should see your custom field on it, as shown below:



Create a request from the customer portal

To test your custom field on a customer request, select Raise a request in the left panel. Select "Desktop /Laptop Support", and check if your custom field is available on the form.

Summary

e.g. My laptop shuts down unexpectedly

Details

Please provide details, such as the model of your laptop (Mac or Windows), or the specific problems you are having (e.g. screen blinks).

Attachment (optional)

Drag and drop files, paste screenshots, or

Dell PowerEdge VRTX

HPE ProLiant DL580 Gen9

Search for an object

That's it for this tutorial

Well done, legend! You're now familiar with the basics of Assets and can easily start working with your objects. If you'd like to dive a bit deeper and see how requests can affect your objects, head on to the next tutorial.

- [Tutorial: How Jira issues affect Assets objects \(ITSM\)](#)

Tutorial: How Jira issues affect Assets objects (ITSM)

Let's dive into how to create a sample IT Service Management (ITSM) project and use it to automate and simplify a typical ITSM problem.

In this tutorial, you'll use the ITSM template, pre-populated with sample data, issue types (incident, problem, change) to see an end-to-end ITSM process. You'll also see how Assets functions can work with Jira workflows and affect objects when a related issue is transitioned through a workflow.

In this tutorial

Here are all the steps you'll follow in this tutorial:

- [1. Create an ITSM project](#)
- [2. Create an incident](#)
- [3. Work on the problem](#)
- [4. Work on the change](#)

1. Create an ITSM project

Let's start by creating a new project based on the Assets' ITSM template, and include some sample data for our tutorial.

- In Jira, the template will create issue types, issue type scheme, workflows, workflow scheme, screens, screen scheme, and issue type screen scheme. Everything you need.
- In Assets, the template will create an object schema, object types with attributes, and object type references.

Since we'll create an ITSM template with sample data, a few Jira issues and Assets objects are also created and linked together.

Create your project

To create a sample project:

1. In the top menu, select **Projects>Create project**.
2. At bottom, select **Create sample data**.
3. Under the **Business** header, select the **Assets IT Service Management** template.
4. Review the issue types and custom fields that will be created and select **Select**.
5. Enter a **Name, Key, and Project Lead** for the project.
6. Select **Submit** to create the project.

View your project's object schema

Your sample project comes together with an object schema in Assets. Here's how to view it:

1. In the top navigation bar, select **Assets**, and select your schema. The name will be similar to the project you created.
2. Feel free to play around with the sample data. Check your object types and their attributes, use the visual graph to see the relations between objects.

The screenshot displays the Jira Assets interface for a project named 'PROJ - ITSM'. On the left, a sidebar (1) shows a tree view of object types such as Business Services, SaaS Services, Hardware, Servers, Computers, Mobile Phones, Server Components, Network Interface, CPU, Memory, Software, Operating System, Applications, Software Licenses, Organization, Employees, Office, Roles, Team, Configuration, and IP Address. The main panel (2) shows the 'Mobile Phones' object type selected, with a list of objects and details for a specific object 'iPhone SN: 12345678'. The details panel includes sections for 'Details', 'Activity', 'Dates', 'Inbound References', 'Attachments', and 'Related Jira issues'. The top navigation bar (3) includes 'Overview', 'Graph', and 'Object Schema' tabs. The 'Object Schema' tab is active, showing the 'Object Type' dropdown (4) and a 'Create Object' button.

1. **Object types:** List of object types in your schema.
2. **Objects:** Selected object type, its child objects, and object details in the main panel.
3. **Object schema actions:** Here you can configure your object schema.
4. **Object type actions:** Here you can configure the selected object type, and switch between viewing its objects, attributes, or the visual graph.

Next up

When you're ready, let's get some work done! Go to [2. Create an incident](#).

2. Create an incident

After you've created your project, you can start working on it right away. Let's imagine we're working as front-line support, and we're reporting an incident.

Create an incident

1. Create an issue in your project, and set its type to **Incident**.
2. Write a summary. Let's say, "I can't send emails".
3. Set the **Affected business service** field to **Email service**. What you're doing here is actually choosing the affected Assets object.

The screenshot shows the 'Create Issue' form with the following fields and values:

- Project***: Mathias ITSM (MI)
- Issue Type***: Incident
- Summary***: I can't send emails
- Affected Business Service**: Email Service
- Reporter***: jeffery

At the bottom right, there are buttons for 'Create another', 'Create', and 'Cancel'.

If you look at the incident you created, you can see that it's linked to an object in Assets. The details of your affected email service automatically appear in the **Affected business service** field.

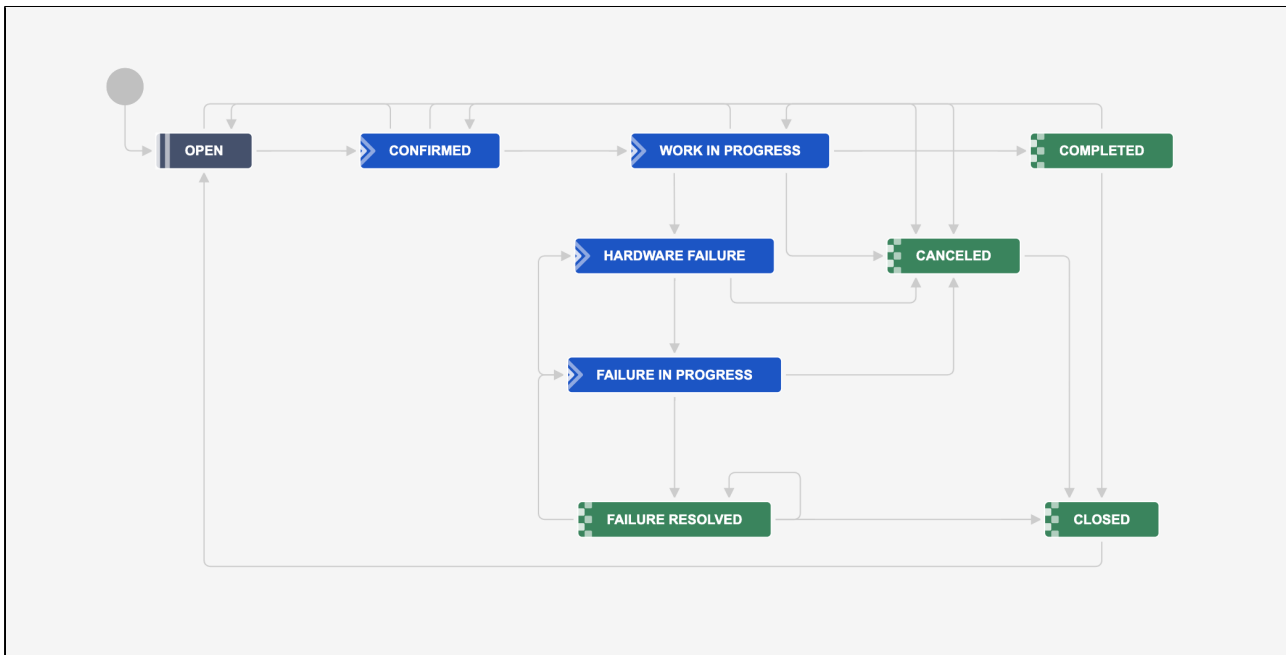
The screenshot shows the 'Details' section for an incident with the following information:

- Type:** Incident
- Priority:** Low
- Labels:** None
- Affected Business Service:** Email Service
 - Status:** RUNNING
 - Business Service Owner:** Wes Shults
 - Service Group Phone:** +46 3284 234 442
 - Importance:** Critical

Additional information shown includes **Status:** OPEN (View Workflow) and **Resolution:** Unresolved.

Related workflow

Here's the workflow that your project uses for incidents, just so you know how the incident will move through it. You can always view it in **Project settings > Workflows**.



Get the work going

Let's transition the incident through it's workflow to see what happens to the Assets objects.

1. Confirm the incident

1. Select **Confirm incident** to confirm it and proceed through the workflow. The status of the affected object changes to Incident in progress.
2. Select **Investigate incident** to start the work. The status of the issue will change.

Details

Type:	Incident	Status:	WORK IN PROGRESS (View Workflow)
Priority:	Low	Resolution:	Unresolved
Labels:	None		
Affected Business Service:	<div style="border: 1px solid gray; padding: 5px; margin-top: 5px;"> <p>Email Service</p> <p>Status: INCIDENT IN PROGRESS</p> <p>Business Service Owner: Wes Shults</p> <p>Service Group Phone: +46 3284 234 442</p> <p>Importance: Critical</p> </div>		

2. Investigate the incident

1. Let's assume that you found the culprit, which is hardware failure. Select **Hardware failure**.

2. Select the **Related Server** that you suspect is the source of the incident and confirm. The current user (you) will be automatically assigned to investigate.

Hardware failure

Summary*

Affected Business Service

Related Server
Related Server for Insight ITSM template

Issue Type* ■ Incident
There are no issue types with compatible field configuration and/or workflow associations. The issue type can only be changed by [moving](#) this issue.

Reporter*
Start typing to get a list of possible matches.

Comment

Style ▾ **B** *I* U A ▾ 🔗 ☺ + ⤴ ⤵ 🔒 Viewable by All Users

An investigation into recent email problems is indicating that an exchange server is intermittently failing.

Hardware failure Cancel

3. Now that you've identified the culprit, the **Related server** from Assets is also linked to the Jira issue. Post-functions built into the template have updated the status of the **Affected business service** and **Related server**.

Details

Type:	■ Incident	Status:	HARDWARE FAILURE
Priority:	↓ Low		(View Workflow)
Labels:	None <small>✎</small>	Resolution:	Unresolved

Affected Business Service:

● Email Service

Status INCIDENT IN PROGRESS

Business Service Owner 👤 Wes Shults

Service Group Phone +46 3284 234 442

Importance Critical

Related Server:

■ exchange.prod.app1

Status INCIDENT IN PROGRESS

Server Owner 👤 Fred Astair

3. Investigate failure

1. Proceed to **Investigate failure**.
2. When enough information has been gathered, resolved the issue with **Resolve**. This indicates that you have determined the incident's cause or that you were unable to reproduce the failure.

3. For the sake of the example, let's assume you can reproduce the incident and have an idea of what's causing it. Change the issue state to **Report problem** to create a new issue of issue type **Problem** to begin tracking the underlying cause of this incident.

Next up

When you're ready to solve some problems, go to [3. Work on the problem](#).


3. Work on the problem

Typically, this is where the role of the front-line support ends. A higher-level support representative might take over the new **Problem** issue, which has its own workflow, reporter and assignee, but includes the information about the affected Assets objects, inherited from the incident.

Where's the problem?

You can see the newly created problem in the issue links of your incident. Select it to see what's it all about.

You should be moved to the problem issue:

Mathias ITSM / MI-19

I can't send emails

Edit Comment Assign More Review Admin

Details

Type: Problem

Status: OPEN View Workflow

Priority: Low

Resolution: Unresolved

Labels: None

Affected Server:

exchange.prod.app1

Status: RUNNING

Environment: Production

Server Owner: Fred Astair

Affected Business Service:

Email Service

Status: RUNNING

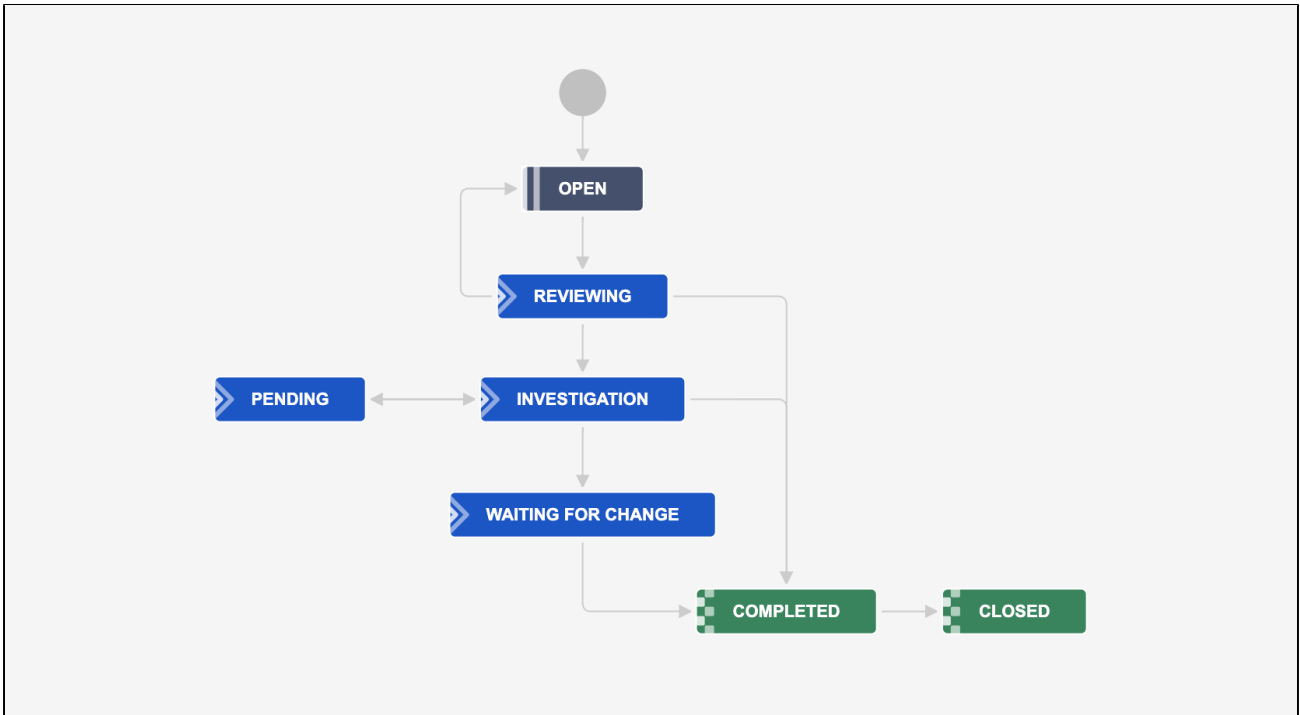
Business Service Owner: Wes Shults

Service Group Phone: +46 3284 234 442

Importance: Critical

Related workflow

Here's the workflow for the Problem issue type, just so you know how your issue will move through it. You can always view it in **Project settings > Workflows**.



Get the work going

1. Investigate the problem

1. Select **Review** and later **Investigate**. This will move your problem right into the investigation state.
2. During the investigation, the assignee could find a component or system that needs to be changed - for example, the problems with the exchange server might be caused by a faulty network card. The assignee would change the ticket status to **REQUEST FOR CHANGE** and select the faulty component.

2. Request for change

1. Select **Request for change**.
2. Select the component that you think is affected from the drop-down list. It will be the job of another person to replace it. In this tutorial, it will be you.

Request for change

Summary* I can't send emails

Affected Server exchange.prod.app1

Affected Server Components Apc Network Management Card 2

Issue Type* 32GB DDR4-2400 ECC REG DIMM

Reporter* jeffery

Start typing to get a list of possible matches.

Comment

- You've selected the faulty component and the status of the issue is now Waiting for change. The affected business service, server, and component are all linked to the issue.

Details

Type: 🚫 Problem

Status: WAITING FOR CHA... [\(View Workflow\)](#)

Priority: ↓ Low

Resolution: Unresolved

Labels: None

Affected Server:

exchange.prod.app1

Status RUNNING

Environment Production

Server Owner Fred Astair

Affected Server Components:

Apc Network Management Card 2

ID APC-2-2

Server [exchange.prod.app 1](#)

Affected Business Service:

Email Service

Status RUNNING

Business Service Owner Wes Shults

Service Group Phone +46 3284 234 442

Importance Critical

Next up

When you're ready to get your hands dirty, go to [4. Work on the change.](#)

4. Work on the change

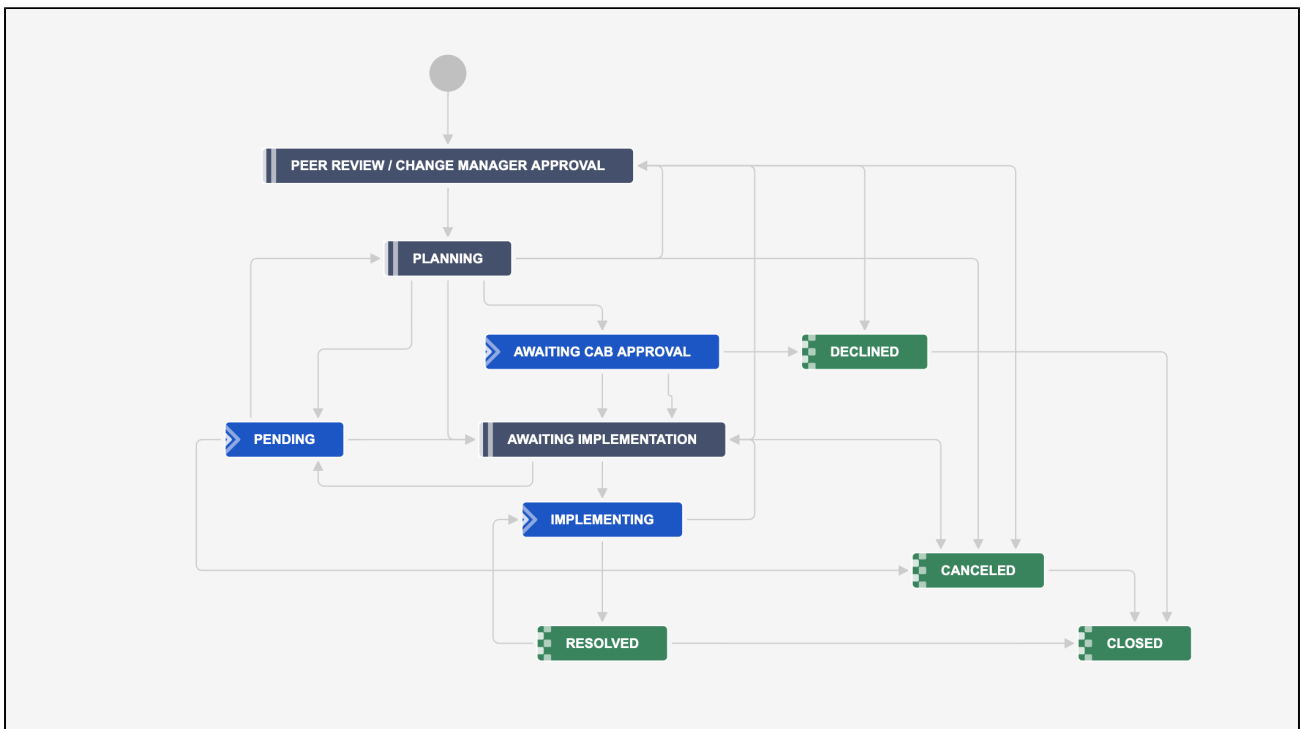
When the Problem issue reaches the Waiting for change status, the new Change issue will be created automatically through Assets automation. Since this ticket requires on-site hardware changes, the support representative managing the Problem ticket might hand this ticket over to the server owner to make the changes.

Where's the change?

Again, you can see the newly created change in the issue links of your problem. Select it to see the details.

Related workflow

Here's the workflow for the Change issue type, just so you know how your issue will move through it. You can always view it in **Project settings > Workflows**.



Get the work going

Approve the change and find a replacement

1. Select **Approve** to approve the change. Here you'll be able to select a replacement component for the malfunctioning server.

- If there are any **Available server components** identified, they will appear in the dropdown list. But, let's say there aren't any and create a new one by selecting **Create an object** in the dialog.

Create Object

Object Type * Network Interface

Name * TP-LINK TG-3468
The name of the object

ID 34366

Server

Status

Network Interface

No description

Avatar

Choose File No file chosen

Create another Create Cancel

- After you've created the object, select it in **Available server component** and progress through the workflow until we reach the **Implementing** state.
- You'll notice when we are implementing the change, the **Affected server** is automatically stopped.

Complete the change

- Select **Change completed**. Here's what happens when you do it:
 - The change ticket is **Resolved**
 - The **Resolution is Done**
 - The faulty server component is changed to **Decommissioned**
 - The **Affected Server** is running again
 - The server component linked to the server is updated to show the new network card
- And above all, the **change** request is resolved.

Mathias ITSM / MI-20

I can't send emails

Edit
 Comment
Assign
More ▾
Delete Me
Close
Admin ▾

▼ **Details**

Type:	Change	Status:	RESOLVED (View Workflow)
Priority:	Low	Resolution:	Done
Labels:	None		
Available Server Components:	<div style="display: flex; align-items: center;"> TP-LINK TG-3468 <div style="margin-left: 20px;"> <p>ID 3434</p> <p>Server exchange.prod.app1</p> </div> </div>		
Affected Server:	<div style="display: flex; align-items: center;"> exchange.prod.app1 <div style="margin-left: 20px;"> <p>Status RUNNING</p> <p>Environment Production</p> <p>Server Owner Fred Astair</p> </div> </div>		
Affected Server Components:	<div style="display: flex; align-items: center;"> Apc Network Management Card 2 <div style="margin-left: 20px;"> <p>ID APC-2-2</p> <p>Status DECOMMISSIONED</p> </div> </div>		

3. Similarly, the **problem** is automatically updated and closed.

The screenshot shows a Jira Service Management problem card. At the top, it displays the user 'Mathias ITSM' and the problem ID 'MI-19'. The title is 'I can't send emails'. Below the title are action buttons: Edit, Comment, Assign, More, Close, and Admin. The 'Details' section is expanded, showing the following information:

- Type: Problem (with a red problem icon)
- Priority: Low (with a green downward arrow icon)
- Status: COMPLETED (in a green box)
- Resolution: Unresolved
- Labels: None (with a blue pencil icon)
- Affected Server: exchange.prod.app1
 - Status: RUNNING (in a green box)
 - Environment: Production
 - Server Owner: Fred Astair
- Affected Server Components: Apc Network Management Card 2
 - ID: APC-2-2
 - Status: DECOMMISSIONED (in a red box)
- Affected Business Service: Email Service
 - Status: RUNNING (in a green box)
 - Business Service Owner: Wes Shults
 - Service Group Phone: +46 3284 234 442
 - Importance: Critical

4. Just like the **incident** that started it all.

This screenshot is identical to the one above, showing the same Jira Service Management problem card for 'I can't send emails' (MI-19) with the same details and status.

That's it for this tutorial

The IT Service Management (ITSM) template is only an example of how you can structure your ITSM practices with Assets and Jira. Feel free to use this template, modify it, or start your own from scratch.

Administering Assets

From Assets roles and permissions to configuring Assets custom fields and functions, and making them work with Jira here you'll find administering tasks that will set up your team for success.

Get started with administering Assets

Here's a list of pages to get you started:

- [Installing Assets](#)
- [Configuring roles and permissions](#)
- [Configuring global Jira settings](#)
- [Configuring global Assets settings](#)
- [Adding Assets custom fields to screens in Jira](#)
- [Enabling Assets custom fields on customer portals](#)
- [Adding Assets functions to workflows in Jira](#)
- [Adding approvers from Assets to requests in Jira](#)
- [Importing and exporting object schemas](#)
- [Assets REST API documentation](#)
- [Advanced usage guides](#)
- [Upgrading Insight with Jira Service Management](#)

Installing Assets

i If you're using Jira Service Management Data Center 4.15 or later, Assets is already included and there's no need to download anything. You can find it by selecting **Assets** in the top Jira navigation.

If you're using Jira Service Management Data Center 4.14 or earlier, you can install the Assets app for free from the [Atlassian Marketplace](#).

Assets is available also for Jira Software Data Center users, but some restrictions apply depending on your Jira version. [See our FAQ for details](#)

Install the app

To install the Assets app from Atlassian Marketplace:

1. Log in to Jira as an admin.
2. Select **Administration > Manage apps**.
3. Select **Find new apps**, and search for **Assets**.
4. The appropriate app version appears in the results.
5. Follow the instructions to install the app. If you're on Data Center, this version won't need any license.
6. You'll be prompted to log into MyAtlassian and Assets will begin to download.
7. Once downloaded, you'll see **Assets** in the top Jira navigation.

System requirements

We strongly believe that your Data Center instance will pull it off, but here's some system requirements, just in case.

Make sure that Jira is configured to handle the amount of data you plan to import into Assets. If you plan to use a lot of scheduling tasks and import a large amount of data we recommend that you follow the table below.

Remember to always test the memory consumption in a test environment for a huge data set, because it's not always the number of objects, but the content of the object attributes as well at play.

Objects in Assets	JVM memory (minimum)	JVM memory (recommended)
~ 1,000,000	16 GB	20 GB
~ 2,000,000	20 GB	25 GB
~ 5,000,000	40 GB	50 GB
~ 6,000,000 + 700k issues with connections	50 GB	60 GB
~ 8,000,000 + 800k issues with connections	70 GB	90 GB

On large installations of Assets, there are some options you can tweak to increase performance.

Garbage collection

To decrease the pause time we recommend that you switch the garbage collection (GC) algorithm to **G1** from the default. You can configure it by adding the following JVM argument to your `JiraSetenvscript`.

```
-XX:+UseG1GC
```

Assets parallelism

Based on your environment, other options to tune the G1 may be required. Check out the information from Oracle (<http://www.oracle.com/technetwork/tutorials/tutorials-1876574.html>) and tune your JVM based on your requirements.

Assets executes tasks in parallel (e.g. reindex, imports). On a large instance, you can increase the number of threads that perform the tasks. You can see this here: [Configuring global Jira settings](#). By default, the parallelism is configured to be equal to the amount of cores available to the JVM.

Database pool

If you have increased the number of threads (Assets parallelism), it is recommended to check the database connection pool monitor and determine if an increased connection pool is required as well.

Jira shutdown

On Jira shutdown Assets persists the index on disk for faster startup. In large instances, the index file fails to be saved on disk in time before the force shutdown command is sent. This will cause Assets to do a database reindex on each startup. To prevent this change, you need to change the timeout on force shutdown in the stop-jira script.

Change the value 20 (the unit is seconds) to a value that is more reasonable based on the amount of data in your installation. Below is a snippet from the standard Jira stop-jira.sh and it is the value **20** that should be changed at two places.

```
if [ -z "$JIRA_USER" ] || [ $(id -un) == "$JIRA_USER" ]; then
    echo executing as current user

    exec $PRGDIR/shutdown.sh 20 -force $@

elif [ $UID -ne 0 ]; then

    echo JIRA has been installed to run as $JIRA_USER so please sudo run this to enable switching to
    that user
    exit 1

else

    echo executing using dedicated user
    if [ -x "/sbin/runuser" ]; then
        sucmd="/sbin/runuser"
    else
        sucmd="su"
    fi
    $sucmd -m $JIRA_USER -c "$PRGDIR/shutdown.sh 20 -force $@"

fi
```

If you have configured your heap with the CATALINA_OPTS in *setenv*, make sure that the JVM configuration (*JVM_MAXIMUM_MEMORY*) is not the same value as the max heap configured in CATALINA_OPTS.

If you are using Assets to accommodate millions of objects and plan to use importers heavily, (e.g. with the Asset Discovery product), then we recommend that you move to Data Center version if you're not already using it. This will offer you the possibility to isolate the importers to only one *import node* with the other nodes being the *user nodes*. With one import node, the import process will have less impact on user interaction. It will also be more robust since any failing import or any over consumption of memory will only affect the import node and not any of your user nodes.

It is difficult for us to set any exact object size or import size recommendation since it's about the data stored on your objects, as well as how often the imports occur. But in general, when reaching millions of objects or memory usage over 32 Gb or if you don't ever want to impact any user interaction, then you should consider using the Data Center version of Assets.

During imports and re-index with Data Center, we need to send messages to other nodes to update the index. This is done through provided Atlassian functionality and the **clustermessagetable** is the DB table where all nodes push and pull. The problem we have seen is that Atlassian have a retention period of **30 days** for this table. We think this is a bit too much and recommend that you should remove all Assets related data rows which are older than 24 hours with your own retention scripts.

Next steps

Head on right into [Administering Assets](#).

Configuring roles and permissions

Learn about roles and permissions available in Assets, and how to grant them to specific users or groups. At the bottom of the page, you can also find a detailed permission matrix that will tell you how permissions correspond to tasks that users can perform.

Roles

A role is a set of permissions granted to Jira users or groups to view or modify data in Assets. Here's a summary of available roles:

Role	Description
Assets Administrator	This role can perform all actions in Assets. This includes: <ul style="list-style-type: none">• Configuring Assets globally• Managing individual schemas
Assets Managers	This role can execute the following tasks on an object schema. This includes: <ul style="list-style-type: none">• Configuring all schema details• Viewing objects• Searching for objects using basic and advanced search• Creating and editing objects• Exporting objects• Printing QR codes and labels
Assets Developers	This role means builders or creators, rather than e.g. software developers. It can execute the following tasks on an object schema: <ul style="list-style-type: none">• Viewing objects• Searching for objects using basic and advanced search• Creating and editing objects• Exporting objects• Printing QR codes and labels
Assets Users	This role can do the following on an object schema: <ul style="list-style-type: none">• Viewing objects• Searching for objects using basic and advanced search• Exporting objects• Printing QR codes and labels

Permission levels


Assets offers three levels of permissions:

Level	Description
Global permissions	<p>These permissions allow you to configure the entire Assets app.</p> <p>By default, as part of Global permissions, the <i>jira-administrators</i> group is mapped to the Assets administrator role. So, if you're a Jira administrator for your Jira instance, then you can perform all actions in Assets.</p> <p>However, if you remove this default mapping, then you wouldn't be able to execute tasks despite of being a Jira administrator.</p>

Object schema permissions	These are permissions to configure and execute actions on the object schema level.
Object type permissions	These are permissions to execute actions on an individual object type. You can view a detailed matrix of all tasks and corresponding permissions at the bottom of this page.

Granting permissions

Assets automatically grants the Assets Administrator role to all Jira admins. However, it might not inherit permissions from any particular active directory running on your Jira instance.

 You need to be a Jira admin to be able to assign users or groups to a specific role in Assets.

Granting global permissions

1. In the top navigation bar, select **Assets > Configure**.
2. Switch to the **Roles** tab.
3. Assign users or groups to the administrator role.

Granting object schema permissions

1. In the top navigation bar, select **Assets > Object schemas**.
2. Open your object schema.
3. In the top-right, select **Object schema > Configure**.
4. Switch to the **Roles** tab.
5. Assign users or groups to the roles.

Granting object type permissions

1. Open your object type.
2. In the top-right, select **Object type > Configure**.
3. Switch to the **Roles** tab.
4. Assign users or groups to the roles.

Good to know

- Object type permissions take precedence over object schema permissions. For example, a user might be assigned to the **User** role on the schema (to not make any changes), but then have a more powerful **Developer** role on a specific object type, so they can create objects.
- If you want users to work with object types (and objects) you need to grant them permissions for object types, but also for the schema (at least **User** permissions, so they can view it). Without giving them any permissions for the schema, they won't be able to access it.
- If you don't specify permissions for object types, they will be inherited from the object schema.

Permission matrix

Here's how roles relate to specific actions users can complete.

Assets functions	Assets Users	Assets Developers	Assets Managers	Assets Administrators	Jira Administrators
Report Jira issues on Assets objects	✓	✓	✓	✓	✓
Search for Assets objects and attributes	✓	✓	✓	✓	✓

View object graph explorer	✓	✓	✓	✓	✓
View object type graph explorer	✓	✓	✓	✓	✓
View Assets custom field	✓	✓	✓	✓	✓
Create/Edit Assets objects		✓	✓	✓	
Move objects (Object Type To) <i>Note that this function applies to roles on the Object type level</i>		✓	✓	✓	
Move objects (Object Type From) <i>Note that this function applies to roles on the Object type level</i>			✓	✓	
Delete Assets objects			✓	✓	
Create/Edit/Delete attributes			✓	✓	
Modify Assets object schema			✓	✓	
Delete Assets object schema				✓	
Create Assets object schema				✓	
Manage icons			✓	✓	
Manage references			✓	✓	
Import object schemas			✓	✓	
Create/Configure/Enable imports			✓	✓	
Manage global Assets permissions				✓	✓
Create/Manage Assets custom field					✓
Move object type (drag & drop)				✓	
Delete object type			✓	✓	
Add/edit Assets post-functions in Jira workflows					✓

Configuring global Jira settings

Global Jira settings for Assets include configuration for the Assets app itself, rather than object schemes, object types, or even objects. Here you can configure log settings, date and time, Assets reindexing, and so on. Read on for more details on available settings.

Accessing global Jira settings

To access global Jira settings for Assets:

1. Go to **Administration > Manage apps**.
2. Look for pages in the Assets section.

General configuration

To open general configuration, select **Assets configuration**.

Setting	Description	
User interaction		
Attribute default label	The text type attribute to be used as default label for every object type. You can change this on the specific object type configuration too. Note that this affects only for those object types that are created after this setting has changed.	No
Attribute default description	The description for the default label attribute. You can change this on the specific object type configuration too.	No
Open Object dialog event	Determines if the object dialog should open when a user selects an object link or when the user hovers with the mouse over an object link. This will also apply when you view Assets object fields while you view a Jira issue.	No

Default number of objects fetched in custom fields	<p>This indicates the number of objects that Assets will fetch in custom fields for each request. The default value is set to 25.</p> <p>Once the user starts to search any objects on the custom field, more objects matching the search criteria are fetched from server asynchronously. Hence, the default limit of 25 is recommended and should be enough.</p> <p>If you increase this number, it will affect the performance since more objects would then have to be fetched on every request.</p>	No
General configuration		
Assets Audit Log enabled	If you check this checkbox, it is ensured that all Assets object events are logged to an audit log file.	No
Include attribute values in audit log	<p>This checkbox is enabled only if the above "Assets Audit Log enabled" is checked.</p> <p>If checked, this will include all attribute values of an object in the audit log.</p>	No
Restore Assets index from file	<p>This will ensure that on start up, Assets index will be restored from a file. This will help increase the startup time.</p> <p>Assets will perform a consistency check of the file against the database on startup and recreate the index if they mismatch.</p> <p>If you uncheck this, you may experience a slower start up time. However, it could remove the risk of a potential corrupted index file which may cause data inconsistency.</p> <p>By default, this is checked. The file is located at <code>{JIRA_HOME}/caches/assets_indexes</code></p> <p>E.g, path of the file on MacOS will be <code>{/var/atlassian/application-data/jira/caches/assets_indexes}</code></p>	No
Store cache on shutdown	<p>This indicates that the Assets index should be persisted to a file on Assets shutdown (e.g, in cases of a plugin upgrade, Jira restart, Assets disable etc).</p> <p>It is recommended that if "Restore Assets index from file" is checked, this property should be switched on too.</p>	No
Restrict Assets Cache	<p>This helps to limit the amount of objects allowed to be stored in Assets. This will subsequently limit the memory footprint by allowing only a limited number of objects to reside in the cache.</p> <p>The default and recommended way to use Assets is to not restrict objects in cache. The limit will have negative performance impact.</p>	Yes
Number of objects allowed in cache	<p>This is enabled only if "Restrict Assets Cache" is checked.</p> <p>This property indicates the number of Assets objects that will be stored in the cache.</p> <p>The recommended way is to not limit objects in cache.</p>	Yes
Maximum File upload size	The maximum file upload size in bytes when uploading files, images, attachments into Assets.	No
Assets parallelism	<p>This is number of threads that Assets will spawn to perform parallel tasks, e.g, importing data, re-indexing etc.</p> <p>If this number is set to a lower value, Assets will put less strain on Jira. However, it will come at the cost of a low performance speed.</p>	No

Process data sources via temp files during imports	Temporarily store data on disk when using the import modules to reduce memory footprint during import.	No
Use a custom locale for Assets	This is used to indicate if the data stored in Assets should be sorted by a locale other than Jira's default one. Fetching of objects may be slower if this option is switched on. Hence, by default, this is disabled to avoid performance issues.	No
The locale for Assets	This is enabled only if "Use a custom locale for Assets" is checked. This determines the language Assets should use when sorting data.	No
Service Desk portal search text (single)	Placeholder for the Assets field on the Jira Service Desk portals (Single fields)	
Service Desk portal search text (multi)	Placeholder for the Assets field on the Jira Service Desk portals (Multiple fields)	
Dedicated Scheduling Node	This node will be the dedicated node to execute Assets scheduling tasks, such as Importers, Automation, etc. Note that, if a node was selected as the dedicated scheduling node and happens to be unavailable at the time of running a scheduled task, then that task will not run.	No

Date settings

All dates in Assets use the Jira administrator settings, and can be changed under the following URL:

```
https://host:port/secure/admin/AdvancedApplicationProperties.jspa
```

Log files

Logs are located in the following directory:

```
<Jira-shared-home>/log
```

Attachments

Assets attachments are stored accordingly in subfolders named avatars, files, icons, and objects in the following directory:

```
<Jira_home>/data/attachments/assets
```

Indexing

To open indexing configuration, select **Indexing Assets**.

ATLASSIAN MARKETPLACE

Find new apps
Manage apps

INSIGHT

Insight Configuration
Indexing Insight
Insight Script Console
Insight reports
Mindville Analytics

Insight Index Information

✔ The Insight index is updated completely. You can still perform a [re-index](#). For more information please see the [Insight Documentation](#).
Note: If you plan to do configuration changes, you may wish to complete those changes before performing the re-index.

Clean re-index Re-index Persist Insight index to file

Index Path: `/var/atlassian/application-data/jira/caches/insight_indexes/objects`

You can select from the following options for indexing:

- Clean re-index**
 A clean re-index means that all objects will be removed from the index, and then will be indexed again. This is recommended if you want to have a fresh state of the index. Once the indexing is in progress, you won't be able to search for objects or filter them.
- Re-index**
 A re-index means that all objects will stay in the index during the process, and Assets will index them again. You can search for objects during the process.
- Persist Assets index to file**
 You can manually persist (copy) the index on your disk. This is useful if you have a big Assets environment with a large number of objects and are planning to reinstall the app. With the index on your disk, Assets won't have to recreate it from scratch.

Testing Groovy scripts

To test Groovy scripts, select **Assets script console**. It gives you a quick and easy way to test Groovy scripts that you want to use in Assets automation or post-functions.

ATLASSIAN MARKETPLACE

Find new apps
Manage apps

INSIGHT

Insight Configuration
Indexing Insight
Insight Script Console
Insight reports
Mindville Analytics

Insight Script Console

This is where you can test groovy scripts that you intend to use in Automation or post functions in Insight.

1

Enter your groovy code in the textarea. Read about examples [here](#)

Issue Key:

Using the variable `issue`, you have access to the issue. Read about it [here](#)

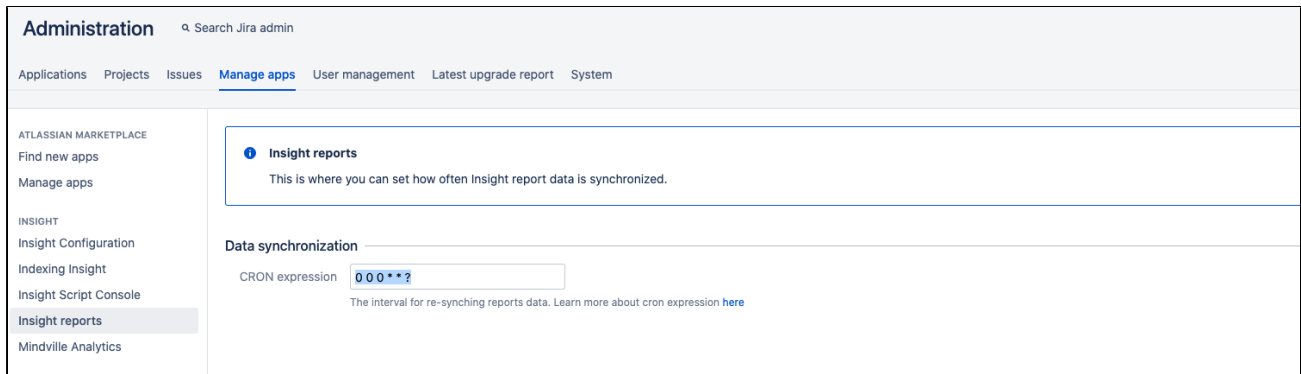
Object Key:

Using the variable `object`, you have access to the object. Read about the `ObjectBean` in our Java API [here](#)

Run

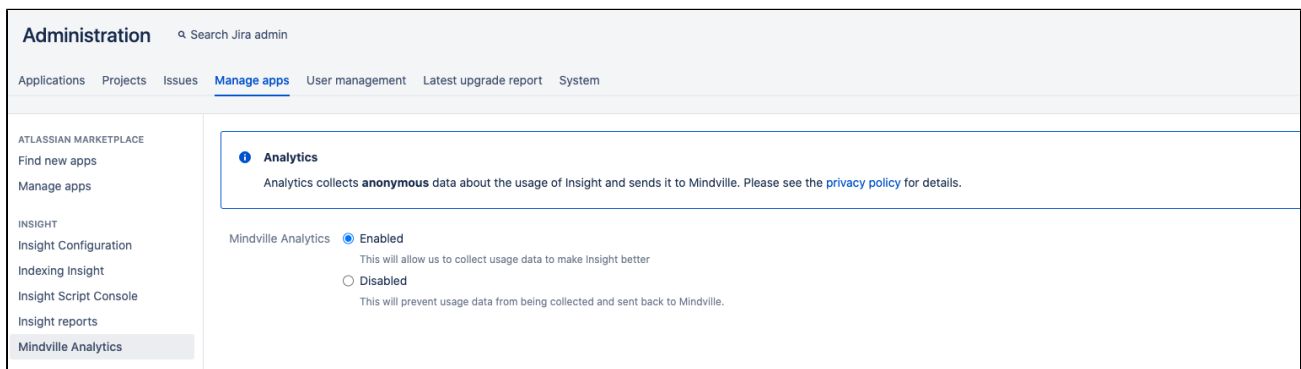
Syncing reports

To open reports syncing configuration, select **Assets reports**. Here you can set up a cron schedule, which syncs the data in your reports.



Analytics

To open analytics configuration, select **Mindville analytics**.



Additional Data Center configuration

Configure data retention period for *clustermessagetable*

Configuring the data retention period helps you avoid performance issues that might result from overloading the *clustermessagetable*. If you import large data sets to Assets in a short period of time, the *clustermessagetable* will be filled with information and can cause performance issues.

To configure the data retention period, complete the following steps:

1. Go to **Administration > System**.
2. Scroll down to the **Advanced** section and select **Services**.
3. Under **Add Service**, under **Class**, select **Build-in services**.
4. Select **Cluster messaging flush service**.
5. Enter the following information:
 - a. **Name**- Cluster Messaging Flush Service
 - b. **Class** -com.atlassian.jira.service.services.cluster.ClusterMessageCleaningService
 - c. **Schedule** -0 0 4/12 * * ?
6. Select **Add Service**.
7. Enter the following for **Retention Period** -2880m
8. Select **Update**.

Services

Name / Class	Properties	Schedule	
Mail Queue Service com.atlassian.jira.service.services.mail.MailQueueService		0 * * * * ?	Edit
Audit log cleaning service com.atlassian.jira.service.services.auditing.AuditLogCleaningService		Daily at 12:00 am	Edit Delete
Backup Service com.atlassian.jira.service.services.export.ExportService	• USE_DEFAULT_DIRECTORY: true	0 0 0/12 * * ?	Edit Delete

Add Service

Add a new service by entering a name and class below. You can then edit it to set properties. Mail handlers should be added or edited via our new [Incoming Mail](#) section.

Name

Class

- ▼ Built-in Services
 - Debugging service
 - Backup service
 - Replicated index flush service
 - Cluster messaging flush service
 - Audit log cleaning service

Schedule Daily
 Days per Week
 Days per Month
 Advanced

Interval at

The current server time is 26/May/20 9:40 AM - Coordinated Universal Time

Configuring global Assets settings

Configuring global Assets settings allows you to choose base references, statuses, and permissions for users. These can be later available when creating object types and objects.

To access global Assets settings:

1. In the top navigation bar, select **Assets > Configure**.
2. Switch tabs to configure the settings you need. You can read more about them below.

References

Objects can have references to other objects, and here is where you create, edit, and delete those references on a global level (additional references can be configured in specific object schemes). You can name a reference, specify an outward and inward description, as well as set a color that will be used by the reference arrows in the object graph.

Name	Description	Color	
<input type="text"/>	<input type="text"/>		Add
Dependency	Dependency		
Financial	Financial		Delete
Link	Link		
Reference	Reference		
Technical	Technical		Delete

Statuses

Objects can have statuses that represent a special attribute type. For example, you can add the attribute **Status** to your object and set it to **Active**. Through automation, this status will change to e.g. **Inactive** when a specific event occurs. Here you can configure global statuses, and set their category to Active, Inactive, or Pending. These statuses will be available to use as attributes in objects.

Good to know:

- You should see the status categories as their state. For example, a new status called **Running** would most likely belong to the **Active** category, while **Broken** to **Inactive**.

References **Statures** Icons Roles Import / Export

i Statures can be set as attributes on objects. You can create, update and delete statures here.

Id	Name	Description	Category	
	<input type="text"/>	<input type="text"/>	■ Active	<input type="button" value="Add"/>
5	Action Needed		■ Active	<input type="button" value="Delete"/>
1	Active		■ Active	<input type="button" value="Delete"/>
7	Closed		■ Inactive	<input type="button" value="Delete"/>
3	In Service		■ Pending	<input type="button" value="Delete"/>
2	Running		■ Active	<input type="button" value="Delete"/>
6	Stopped		■ Inactive	<input type="button" value="Delete"/>
4	Support Requested		■ Pending	<input type="button" value="Delete"/>

Global permissions

As a member of the Jira-administrators group, you can specify who should have administrative permissions in Assets. Based on the role users have (User, Manager, Developer, Administrator), they might only be able to view existing objects, or also create new objects, object types, and even object schemes. It really depends on how your company works; probably there will be a set of people responsible for creating objects, while the rest can only view them and raise requests against them.

For more info about what each role can do, see [Roles and permissions](#).

Importing and exporting object schemas

Here you can import an existing object schema or export that one you're using. This is useful if you'd like to create a backup or have an existing schema in a different instance of Assets.

References **Statures** Icons Roles **Import / Export**

Available Imports

Object Schema Import

You can import a new object schema from the server.

ENABLED

Object Schema Export

You can export an existing object schema to a file on the server.

ENABLED

i Notice!

There are several more import types within each object schema configuration!

For more info, see [Importing and exporting object schemas](#).

Adding Assets custom fields to screens in Jira

You can have your objects displayed in Jira issues by adding Assets custom fields to Jira screens (just like you would with your usual Jira custom fields). This could be adding the affected business service to your incident issues, adding a computer to a hardware request issue, or adding a set of potentially affected hosts to a change request issue.

The screenshot shows the 'Details' section of a Jira issue. The issue type is 'Problem' and its status is 'OPEN' (with a 'View Workflow' link). The resolution is 'Unresolved'. There are no labels. The 'Affected Server' is 'exchange.prod.app1', which is 'RUNNING' in the 'Production' environment, owned by 'Fred Astair'. The 'Affected Business Service' is 'Email Service', which is 'RUNNING', owned by 'Wes Shults', with a service group phone of '+46 3284 234 442' and an importance of 'Critical'.

Quick overview

Here's a video on Assets custom fields if you'd like get a quick overview:

Creating Assets custom fields

You can create three types of custom fields: Default, Referenced, and Readonly.

- [Default Assets custom field](#)
- [Referenced Assets custom field](#)
- [Read-only Assets custom field](#)

Enabling access for customers

When you add Assets custom fields to Jira Service Management, you need to enable them for customer portals. Without these steps, customers won't be able to access these fields. [Learn more about enabling Assets custom fields on customer portals](#)

Default Assets custom field

The default Assets custom fields gives you the most flexibility, and is what you'll use in most cases. When you're looking to add objects to issues while creating them, the default custom fields will help you achieve that.

Create a default custom field

To create the default Assets custom field:

1. Log in to Jira as an admin.
2. Go to **Administration > Issues > Custom fields**.
3. Select **Add custom field > Advanced**.
4. Search for **Assets object/s**, and select it.
5. Enter a name and description for your field.
6. Once created, associate the custom field with desired screens in Jira.

The Assets default custom field is now created and associated with the right screen in Jira. Now, you need to configure this field so it allows your users to choose specific objects.

Configure the default custom field

With your custom field created and associated with the right screens in Jira, you can now configure how it will behave.

1. Find your custom field on the Custom fields page, and select **Configure** next to it.
2. Select **Edit Assets configuration**.
3. Select which object schema the custom field should use.
4. Configure other settings. You can check what they mean in the table below.

Scope

Object Schema	PROJ - ITSM	i
Filter Scope (AQL)	objectType in objectTypeAndChildren("Server Components")	i
Filter Issue Scope (AQL)	Server IN (\${Affected Server\${0}})	i
Filter Assign Scope (AQL)	Not configured	i
Make Filter Assign Strict	<input type="checkbox"/>	i

User Interaction

Type of interaction	<input checked="" type="radio"/> Select <input type="radio"/> Checkbox / Radio button	i
Filter objects with attributes	Label x	i
Object attributes on Issue view	Server x ID x Status x	i
Multiple	<input type="checkbox"/>	i
Option to add all objects	<input type="checkbox"/>	i

Jira Service Management

Show object picker on customer portals	<input checked="" type="checkbox"/>	i
Force to show on Customer Issue details	<input type="checkbox"/>	i

[Return to Jira Custom Field](#)

Scope

Setting	Description
---------	-------------

Object schema	Object schema that the custom field should use. If you want this field to refer to specific objects, make sure you point to the right schema.
Filter scope (AQL)	Define the scope of objects that are valid for this field. For example, if you want it to show only specific object types (Applications, but not Hardware), then you could use a query like this one: <code>objecttype = Application</code>
Filter issue scope (AQL)	In the create/edit view, and in the customer portal, you can filter objects based on their attributes, in addition to the filter scope you configured above. This doesn't apply to JQL searches. Placeholders are supported, but only for "Select type". Example This AQL will show objects that are owned by the group selected in the request: <code>Server."Owner Group" = \${Group}</code> This AQL will show objects that are owned by the reporter: <code>Owner = currentReporter()</code>
Filter assign scope (AQL)	Allows you to automatically set objects to the custom field. This can be filtered on any issue field values with support for placeholders. Example This AQL will show objects that are owned by the group selected in the request: <code>Server."Owner Group" = \${Group}</code> This AQL will show objects that are owned by the reporter: <code>Owner = currentReporter()</code>

For more info on creating your queries, see:

- [AQL - Assets Query Language](#)
- [Advanced: Placeholders](#)

User interaction

Setting	Description
Type of interaction	You can specify what type of interaction the custom field should use: <ul style="list-style-type: none"> • Select • Checkbox / Radio buttons
Filter objects with attribute	You can specify the attributes to filter on, this field is mandatory and at least one has to be configured. Label is default.
Object attributes on issue view	You can specify which of the object's attributes will be displayed in a Jira issue. For example, you might specify only the most important attributes, so as to not clutter your issues, and users can view the rest after opening the object.
Multiple	You can specify if users can choose multiple values for this field.

Option to add all objects	If you allowed multiple values (above), you can add an option to select all values.
----------------------------------	---

Service Desk

Setting	Description
Show object picker on customer portals	This is for Jira Service Management. You can select whether the object picker (which allows to browse and select objects) should be available on the customer portal.

Enable access for customers

When you add Assets custom fields to Jira Service Management, you need to enable them for customer portals. Without these steps, customers won't be able to access these fields. [Learn more about enabling Assets custom fields on customer portals](#)

Referenced Assets custom field

A referenced Assets custom field allows you to specify a parent custom field and then shows objects that have references to or from the object chosen in the parent field.

Example

Parent field: Affected server	Reference	Referenced field: Affected part
Biggest server in the server room	CPU	Intel(R) Xeon(R) CPU E7- 4830 @ 2.13GHz
	Network interface	Apc Network Management Card 2
	Memory	32GB DDR4-2400 ECC REG DIMM

The objects available in your referenced custom field will change depending on what the user has chosen in the parent custom field.

Creating a referenced custom field

To create a referenced custom field:

1. Log in as the Jira administrator.
2. Go to **Administration > Issues > Custom fields**.
3. Select **Add custom field**, and select **Advanced**.
4. Search for **Assets referenced object**, and select either the single- or multi-select field.
5. Enter a name and description for your field.
6. Associate the custom field with desired screens in Jira.

Configuring a referenced custom field

With your custom field created and associated with the right screens in Jira, you can now configure how it will behave.

1. Find your custom field on the Custom fields page, and select **Configure** next to it.
2. Select **Edit Assets configuration**.
3. Select which object schema the custom field should use.
4. Configure other settings. You can check what they mean in the table below.

Configure Assets Custom Field - Affected Server (customfield_10200)

Scope

Parent Custom Field	Affected Server	i
Reference direction	Outbound references	i
Reference type	<input checked="" type="checkbox"/> Reference	i
Restrict to User related objects	<input type="checkbox"/>	i
Restrict to Project related objects	<input type="checkbox"/>	i
Additional Filter Scope (AQL)	Order By Label	i
Filter Issue Scope (AQL)	Not configured	i

User Interaction

Filter objects with attributes	<input type="text" value="Label x"/>	i
Object attributes on Issue view	<input type="text" value="Version x"/>	i
Option to add all objects	<input type="checkbox"/>	i

Jira Service Management

Show object picker on customer portals	<input checked="" type="checkbox"/>	i
Force to show on Customer Issue details	<input type="checkbox"/>	i

[Return to Jira Custom Field](#)

Setting	Description
Scope	
Parent custom field	Select the parent custom field. Your referenced custom field will return objects based on what was chosen in the parent custom field.
Reference direction	Select the direction of the references: <ul style="list-style-type: none"> ▪ Outbound references: All references that exist on the object chosen in the parent custom field. This would typically include all attributes of type Objects, which means references to other Assets objects. ▪ Inbound references: All references from other objects that go to the object chosen in the parent custom field.
Reference type	Reference type. You can choose any type that you have configured in Assets or leave it blank to use all references.
Restrict to user related object	You can restrict this custom field to just show objects related to the Jira users filling in data for this field. You can additionally restrict it to a specific user by using AQL. For more info on creating your queries, see AQL - Assets Query Language .
Restrict to project related objects	You can restrict this custom field to just show objects related to a project where this field is used.

Additional filter scope (AQL)	<p>You can filter objects on attributes to select only those that fit your needs.</p> <p>An example:</p> <pre>Status = Active</pre> <p>For more info on creating your queries, see AQL - Assets Query Language.</p>
Filter issue scope (AQL)	<p>In the create/edit view, and in the customer portal, you can filter objects based on their attributes, in addition to the filter scope you configured above. This doesn't apply to JQL searches. Placeholders are supported, but only for "Select type".</p> <p>Here's an example:</p> <pre>"System Owner".User = currentReporter()</pre> <p>For more info on creating your queries, see AQL - Assets Query Language.</p>
User interaction	
Filter objects with attributes	<p>You can specify the attributes to filter on, this field is mandatory and at least one has to be configured. Label is default.</p>
Object attributes on issue view	<p>You can specify which of the object's attributes will be displayed in a Jira issue. For example, you might specify only the most important attributes, so as to not clutter your issues, and users can view the rest after opening the object.</p>
Jira Service Management	
Show object picker on customer portals	<p>This is for Jira Service Management. You can choose whether the object picker (which allows to browse and select objects) should be available on the customer portal.</p>
Force to show on Customer Issue details	<p>This is for Jira Service Management. You can choose whether the custom field is visible in customer issue details, even if it's not a visible request field.</p>

Enabling access for customers

When you add Assets custom fields to Jira Service Management, you need to enable them for customer portals. Without these steps, customers won't be able to access these fields. [Learn more about enabling Assets custom fields on customer portals](#)

Read-only Assets custom field

The readonly custom field allows you to include information about objects in your issues, without being able to change this information, as it's read only.

Example

An example is an employee off-boarding you can use this field to display the employee's current assets on the off-boarding issue, so you know what needs to be collected before they leave.

Creating a read-only custom field

1. Log in as the Jira administrator.
2. Go to **Administration > Issues > Custom fields**.
3. Select **Add custom field**, and select **Advanced**.
4. Search for **Assets readonly object**, and select it.
5. Enter a name and description for your field.
6. Associate the custom field with desired screens in Jira.

Configuring the read-only custom field

With your custom field created and associated with the right screens in Jira, you can now configure how it will behave.

1. Find your custom field on the Custom fields page, and select **Configure** next to it.
2. Select **Edit Assets configuration**.
3. Select which object schema the custom field should use.
4. Configure other settings. You can check what they mean in the table below.

Configure Assets Custom Field - CPU (customfield_10202)

Scope

Object Schema	Resources	i
Filter Scope (AQL)	objectType = "Computer"	i
Filter Assign Scope (AQL)	Not configured	i

User Interaction

Object attributes on Issue view Manufacturer x Display Size x ▼ [i](#)

[Return to Jira Custom Field](#)

Setting	Description
Scope	
Object schema	Object schema that the custom field should use. If you want this field to refer to specific objects, make sure you point to the right schema.
Filter scope (AQL)	This is the "core" scope for this field. Any searching in Jira (JQL) will apply to this AQL. <pre>objectType = "Computer"</pre>

Filter assign scope (AQL)	<p>Allows you to automatically set objects to the custom field. This can be filtered on any issue field values with support for placeholders.</p> <pre data-bbox="411 230 1428 376"> //Assign objects depending on the reporter Location.Employee = currentReporter() //Can be specified like this as well due to support for place holders Location.Employee = \${reporter} </pre>
User interaction	
Object attributes on issue view	You can specify the attributes that should be shown on any object attached to an issue in the Jira issue view.

Enabling access for customers

When you add Assets custom fields to Jira Service Management, you need to enable them for customer portals. Without these steps, customers won't be able to access these fields. [Learn more about enabling Assets custom fields on customer portals](#)

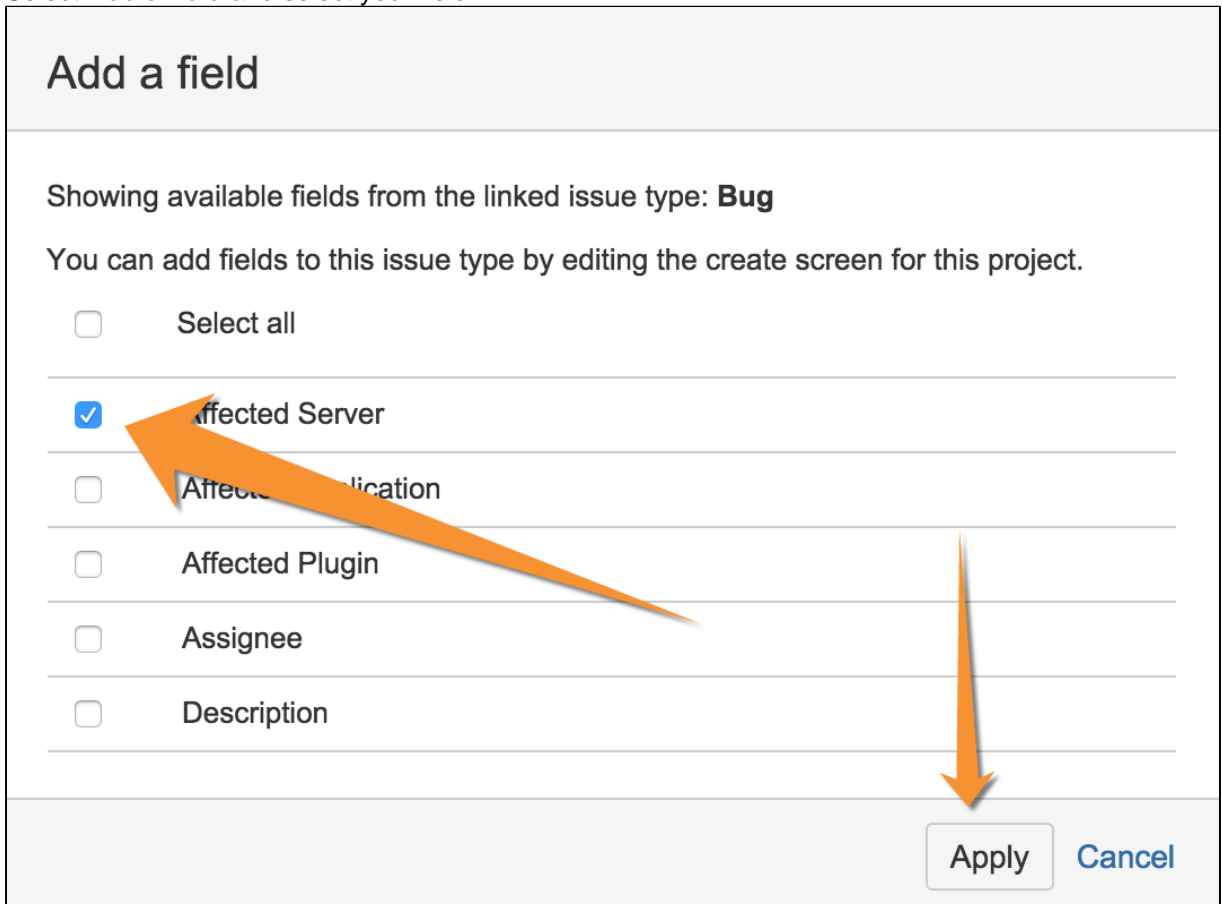
Enabling Assets custom fields on customer portals

When you add Assets custom fields to Jira Service Management, you need to enable them for customer portals. Without these steps, customers won't be able to use these fields.

Enable custom fields on customer portals

Add custom field to your service project

1. Go to your service project.
2. Go to **Project settings > Customer portal**.
3. Select the request type to which you want to add your custom field.
4. Select **Edit form**, and switch to the **Fields** tab.
5. Select **Add a field** and select your field.



Add a field

Showing available fields from the linked issue type: **Bug**

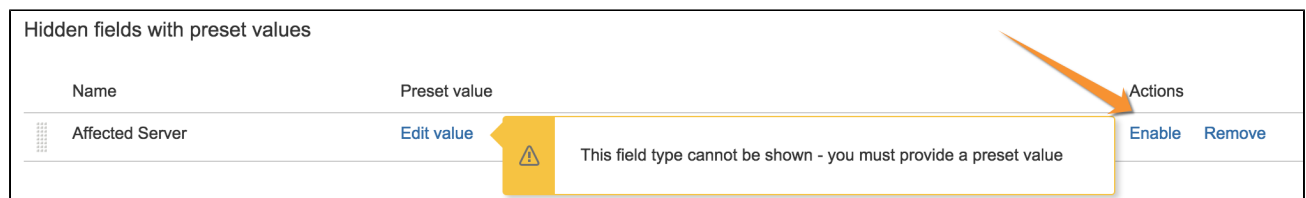
You can add fields to this issue type by editing the create screen for this project.

- Select all
- Affected Server
- Affected Application
- Affected Plugin
- Assignee
- Description

[Apply](#) [Cancel](#)

Enable and show the custom field

Enable the custom field:



Name	Preset value	Actions
Affected Server	Edit value	Enable Remove

This field type cannot be shown - you must provide a preset value

Show the custom field:

Hidden fields with preset values		
Name	Preset value	Actions
Affected Server	Edit value	Show Remove

You must provide a value when a field is hidden.

Enable access for customers

Once your custom field is enabled, you need to configure access on your object schema so that customers can select objects in your field:

1. Open your object schema in Assets.
2. At the top-right, select **Object schema > Configure**.
3. Switch to the **Roles** tab.
4. In the Jira Service Management tooltip, select **Enable**.

Configure Asset Demo

General
References
Statuses
Icons
Roles
Import

i

The following roles have permission to manage and view this object schema. Users can only view the object schema, Developers are able to create/edit object modify.

Roles	Users	Groups
Object Schema Managers		jira-developers
Object Schema Users		jira-users
Object Schema Developers		jira-developers

i

JIRA Service Desk

Object Schema permissions for Service Desk Customers are DISABLED

[Enable](#)

Adding Assets functions to workflows in Jira

Assets includes several post functions, conditions and validators, and Groovy scripts that let you automate the execution of actions. By adding to workflows in Jira, you can trigger actions on your objects when an issue transitions through a workflow. These actions include updating an asset, sending a notification, or running a script.

For instance, when an issue is created requesting the onboarding of an employee, you can assign necessary assets to the new user including laptop, cell phone and cell phone subscription.

Get started with Assets functions

- [Conditions and validators](#)
- [Post functions](#)
- [Groovy scripts](#)

Conditions and validators

Conditions and validators allow you to add certain requirements on workflow transitions to make sure that your request can be transitioned only if these requirements are met. Insight introduces conditions and validators that are related to object attributes (for example, you'd allow a change request for a server only if it's not in production).

To view available conditions and validators:

1. Open a workflow that you'd like to add the conditions or validators to.
2. Select a transition. Conditions and validators are added to a transition rather than a workflow step.
3. Select **Conditions** or **Validators**, then select **Add**.
4. Select **Assets condition** or **Assets validator**.
5. Select one of the available options. Learn more about them in the table below.

Available options

Here are the Assets conditions or validators that you can use:

Name	Type	Description
Validate on object attributes	Condition	Allows to transition a request only if the related object's attributes meet this condition/validator's criteria. For example:
	Validator	<ul style="list-style-type: none">▪ Allow change request only if the target server has enough licenses available▪ Allow change request only if the target server is not in production▪ Allow holiday request only if the user has enough remaining vacation days▪ Allow specific hardware request to be approved only if there's enough hardware models available▪ Allow only specific requests on server that have a sufficient SLA level <p>You'd build your queries here using AQL - Assets Query Language.</p>
Run a Groovy script	Condition	Allows you to run a groovy script within the conditions/validator, which can be useful when you need some specific requirements met.
	Validator	For more info about Groovy scripts and examples, see Groovy scripts .

Post functions

Adding post functions to your Jira workflows allow you to trigger actions against your objects when an issue is transitioned through the workflow. Here you can find available post functions for Assets.

To view available conditions and validators:

1. Open a workflow that you'd like to add the post functions to.
2. Select a transition. Post functions are added to a transition rather than a workflow step.
3. Select **Post-functions**, then select **Add**.
4. Select **Assets Post-function**.
5. Select one of the available options. Learn more about them in the table below.

Available options

Here you can find the available post functions.

The selected object will be added, removed, or set to the corresponding attribute of the target object.

Name	Description
Condition	Specify the Groovy code to use as condition for this post function to execute. Condition ignored on empty value.
Source custom field	Specify the custom field where objects are selected and to be added, removed, or set.
Change type	Specify the change type.
Target custom field	Specify the custom field where objects should be changed.

The selected user custom field will be used to find Assets objects that are associated with that user.

Name	Description
Condition	Specify the Groovy code to use as condition for this post function to execute. Condition ignored on empty value.
Source custom field	Specify the custom field where information (email addresses or Jira users) can be found.
Target custom field	Specify the custom field where objects should be assigned.
Assets object type attribute name	Optional object type attribute to filter objects on.

Select an object attribute value of a selected object and set it to another target custom field. Supported fields are Jira Text fields, Jira user fields, Jira group fields and Assets custom fields (not reference fields).

Name	Description
Condition	Specify the Groovy code to use as condition for this post function to execute. Condition ignored on empty value.
Source custom field	Specify the Assets custom field from where objects attribute should be fetched.
Assets object type attribute name	Specify the name of the object type attribute where to fetch the value.

Target custom field	Specify the custom field where object attribute should be stored (reference fields can't be selected).
----------------------------	--

Select an Assets object and specify the attribute the assignee will be fetched from.

Name	Description
Condition	Specify the Groovy code to use as condition for this post function to execute. Condition ignored on empty value.
Source custom field	Specify the Assets custom field from where objects attribute should be fetched.
Assets object type attribute name	Specify the name of the object type attribute the user will be fetched from. Any type except "User" or "Default / Email" will be ignored.

Select an Assets object and specify the "User" or "Default / Email" attribute to which an email will be sent.

Name	Description
Condition	Specify the Groovy code to use as condition for this post function to execute. Condition ignored on empty value.
Source custom field	Specify the Assets custom field from where objects attribute should be fetched.
Assets object type attribute name	Specify the name of the object type attribute the user will be fetched from. Any type except "User" or "Default / Email" will be ignored.
Email subject	Specify the email subject. Issue placeholders are supported.
Email message	Specify the email message. Issue placeholders are supported.

Increase or decrease an object attribute value (of type Integer) by a set amount.

Name	Description
Condition	Specify the Groovy code to use as condition for this post function to execute. Condition ignored on empty value.
Source custom field	Specify the Assets custom field from where objects attribute should be fetched.
Assets object type attribute name	Specify the name of the object type attribute the user will be fetched from. Any type except "User" or "Default / Email" will be ignored.
Change type	Specify the change type: Increase or Decrease.
Change value	Specify the change value. Anything except a number will be set to default (1).

Selected objects will have their specified attribute set to the predetermined value.

Name	Description
Condition	Specify the Groovy code to use as condition for this post function to execute. Condition ignored on empty value.
Source custom field	Specify the Assets custom field from where objects attribute should be fetched.

Assets object type attribute name	Specify the name of the object type attribute where to set the value.
Change value	Specify the value to set. Any value not validated to the attribute will be ignored. Issue placeholders are supported. Leaving this field empty will set an empty value.

Write a Groovy script when you have some specific requirements that the predefined Assets post functions can't handle. About parameters into the Groovy engine, you have all the parameters available as in a normal post function. The most common one is "issue" where you may query as you like. Check out Assets [Groovy script examples](#).

Name	Description
Groovy script	Enter your Groovy code.

Select an Assets object schema, object type, and configure the attributes to be populated from a Jira Issue.

⚠ The **value** of the Issue field mapped to an object attribute must be **data format** compatible, otherwise Assets won't be able to create a new object. An example of a mapping that will fail: A **Jira URL picker** mapped to an object attribute of type **Integer**.

Name	Description
Condition	Specify the Groovy code to use as condition for this post function to execute. Condition ignored on empty value.
Assign to issue	The newly created object isn't automatically linked to the issue. To assign it to the issue, select the custom field on the issue to contain the reference.
Always create	If selected, Assets will create the object even if mandatory object attributes haven't been mapped correctly.
Object schema	Specify the object schema that contains the object type of interest.
Object type	Specify the object type that corresponds to the type of object that should be created.
Attributes/ Issue fields	Once the object type is selected, the list of attributes will be listed, allowing a mapping against an Issue field.

Supported Jira issue fields:

Field	Jira type	Object type attribute type	Expected object type attribute value
Affected version (s)	Version (s)	<ul style="list-style-type: none"> Version Numeric Other 	<ul style="list-style-type: none"> Jira Version(s) The Version ID(s) Jira Version name(s)
Assignee	User	<ul style="list-style-type: none"> User Email Other 	<ul style="list-style-type: none"> Jira User Jira User email Jira User display name

Description	Text	<ul style="list-style-type: none"> • Text area • Other 	<ul style="list-style-type: none"> • Description in Wiki format • Description
Due date	Date	<ul style="list-style-type: none"> • Date • Date time • Other 	<ul style="list-style-type: none"> • Date • Date time • System/User locale format of Date
Fix version(s)	Version (s)	<ul style="list-style-type: none"> • Version • Numeric • Other 	<ul style="list-style-type: none"> • Jira Version(s) • Jira Version ID(s) • Jira Version name(s)
Issue type	Issue type	<ul style="list-style-type: none"> • Numeric • Other 	<ul style="list-style-type: none"> • Issue type ID • Issue type name
Key (issue)	Text	<ul style="list-style-type: none"> • Any 	<ul style="list-style-type: none"> • The Issue key
Labels	Label(s)	<ul style="list-style-type: none"> • Any 	<ul style="list-style-type: none"> • The label(s) value(s)
Priority	Priority	<ul style="list-style-type: none"> • Numeric • Other 	<ul style="list-style-type: none"> • Priority ID • Priority name
Project	Project	<ul style="list-style-type: none"> • Project • Numeric • Other 	<ul style="list-style-type: none"> • Jira Project • Jira Project ID • Jira Project name
Reporter	User	<ul style="list-style-type: none"> • User • Email • Other 	<ul style="list-style-type: none"> • Jira User • Jira User email • Jira User display name
Resolution	Resolution	<ul style="list-style-type: none"> • Date • Date time • Other 	<ul style="list-style-type: none"> • Resolution Date • Resolution Date time • Resolution name
Resolved	Date	<ul style="list-style-type: none"> • Date • Date time • Other 	<ul style="list-style-type: none"> • Date • Date time • System/User locale format of Date
Status (Issue)	Status	<ul style="list-style-type: none"> • Any 	<ul style="list-style-type: none"> • Status name
Summary	Text	<ul style="list-style-type: none"> • Any 	<ul style="list-style-type: none"> • The summary

customfield_XXXXX	Undefined	<ul style="list-style-type: none"> Any 	The value of the custom field.
--------------------------	-----------	---	--------------------------------

Make sure that when using custom fields, the custom field value is possible to set to the target attribute. Otherwise, it will be discarded.

Assign objects to custom fields depending on a configured AQL query. Any issue value is available as place holders. Use the common field definitions for Jira fields. The AQL query will run upon the schema which the Assets custom field belongs.


Name	Description
Condition	Specify the Groovy code to use as condition for this post function to execute. Condition ignored on empty value.
AQL	Specify the AQL to filter the objects to set on the specified custom field in next configuration. The AQL supports placeholders from any issue field.
Target custom field	Specify the Assets custom field that the filtered objects should be attached. (Referenced Assets Custom fields aren't supported)

Groovy scripts

You can run Groovy scripts both in [conditions and validators](#) and [post functions](#). They give you more power and flexibility, but are recommended for advanced users.

Good to know

Each Groovy script has to be whitelisted. To add Groovy scripts to the allowlist:

1. Go to **Administration**  > **Manage apps** > **Assets Allowlist**.
2. Select **Edit settings**.
3. Enter the filename and the full path to the script like in the example below:

Allowlist Configuration
Here is where you configure general allowlist configuration for Assets

Automation

Allowlisted groovy scripts: `/Users/jira/jira-home/atlassian-jira-servicedesk-4.20.0/scripts/Move Object to Archive.groovy`

[Edit Settings](#)

4. Select **Save**.

Ensure that the directory and the subdirectories where your Groovy script is located are readable by all users and executable by the user running Jira. For example:

```
= $ ls -al ~/Users/jira/jira-home/atlassian-jira-servicedesk-4.20.0/scripts
total 8
drwxr-x---  3 jira  staff   96 Nov 24 21:47 .
drwxr-xr-x 21 jira  staff  672 Dec  1 15:48 ..
-rw-r--r--  1 jira  staff 2421 Nov 24 21:47 Move Object to Archive.groovy
```

Variables

When adding a Groovy script, the Assets workflow functions has the following variables pre-defined, so you can use them in your scripts.


Variable	Description
issue	The issue to be transitioned.
originalIssue	The original issue. You can check conditions, like if the assignee has changed by: <pre>issue.assignee != originalIssue.assignee</pre>
currentUser	The current Jira user. You can check if the reporter is the current user by: <pre>currentUser == issue.reporter</pre>

log	A log framework. You can log to <code>atlassian-jira.log</code> by: <pre>log.warn("This is a warning!");</pre>
------------	---

Examples

If you're looking for some inspiration and examples, see [Groovy script examples](#).

Adding approvers from Assets to requests in Jira

 You need to have a Data Center license to use this feature.

Jira Service Management allows you to require approvals for a workflow step. This means that a request can't move from one status to another until it's approved by the right people. In Assets, the right people are the owners of your assets or groups responsible for their maintenance. You can now add them as approvers when a request is raised against the asset they own or have relations to. The approvers will change dynamically based on the assets selected in your requests.

Before you begin

To add approvers from Assets, you'll need the following items:

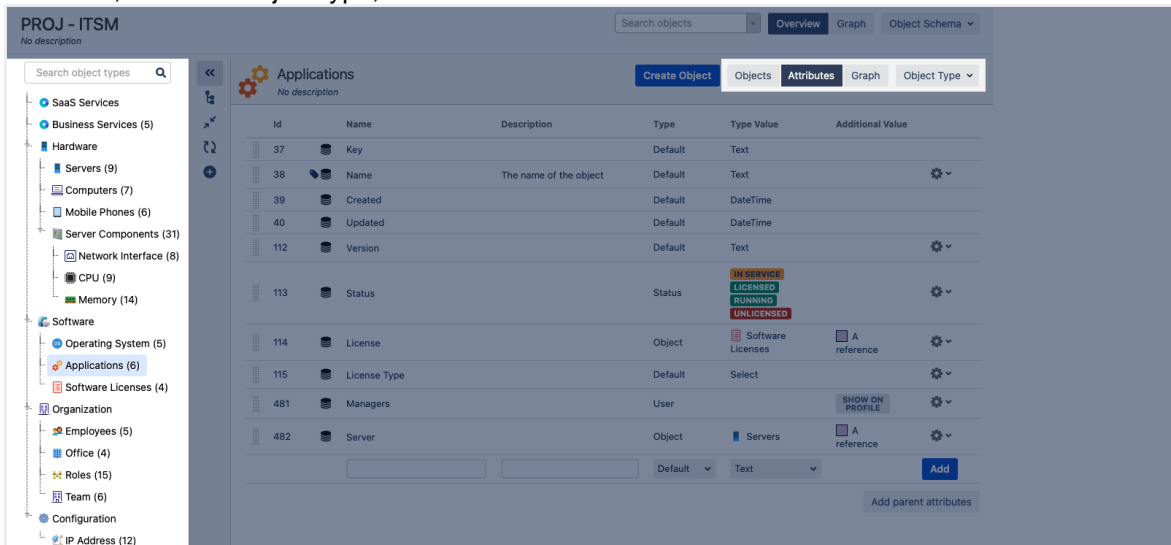
- Attribute of type User or Group added to an object type. If you select groups, keep them rather small as bigger groups will affect performance.
- One of the Assets object custom fields mapped to your objects in Assets. Supported types are:
 - Assets objects
 - Assets object (single) (deprecated)
 - Assets object (multiple) (deprecated)

If you don't have them already, follow these steps:

When you add the User/Group attribute to your object type, you can then fill it in for the underlying objects by choosing a specific user or group. These people will have relations to your objects and can be assigned as approvers. For more info on attributes, see [Adding attributes to object types](#).

To add an attribute to your object type:

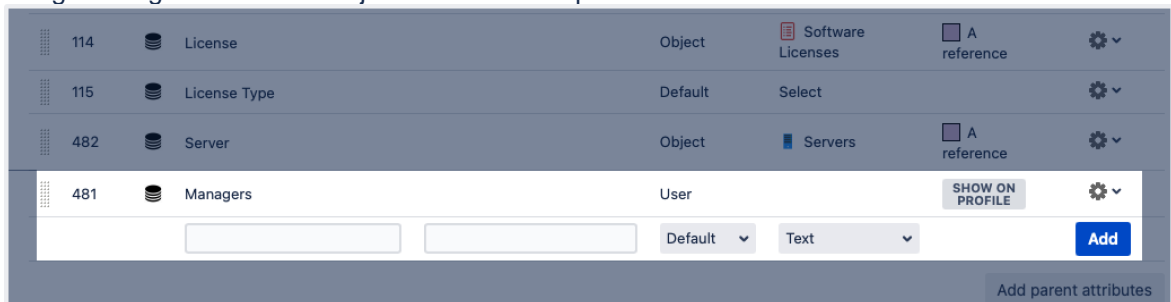
1. In Assets, select an object type, and switch to the **Attributes** tab.



The screenshot shows the 'Attributes' tab for the 'Applications' object type in Jira Assets. The interface includes a search bar, a 'Create Object' button, and tabs for 'Objects', 'Attributes', 'Graph', and 'Object Type'. A table lists attributes with columns for Id, Name, Description, Type, Type Value, and Additional Value. The 'Managers' attribute (Id: 481) is highlighted, showing it is of type 'User' and has a 'SHOW ON PROFILE' checkbox. Below the table, there are input fields and dropdown menus for adding a new attribute, with an 'Add' button.

Id	Name	Description	Type	Type Value	Additional Value
37	Key		Default	Text	
38	Name	The name of the object	Default	Text	
39	Created		Default	DateTime	
40	Updated		Default	DateTime	
112	Version		Default	Text	
113	Status		Status	IN SERVICE LICENSED RUNNING UNLICENSED	
114	License		Object	Software Licenses	A reference
115	License Type		Default	Select	
481	Managers		User		SHOW ON PROFILE
482	Server		Object	Servers	A reference

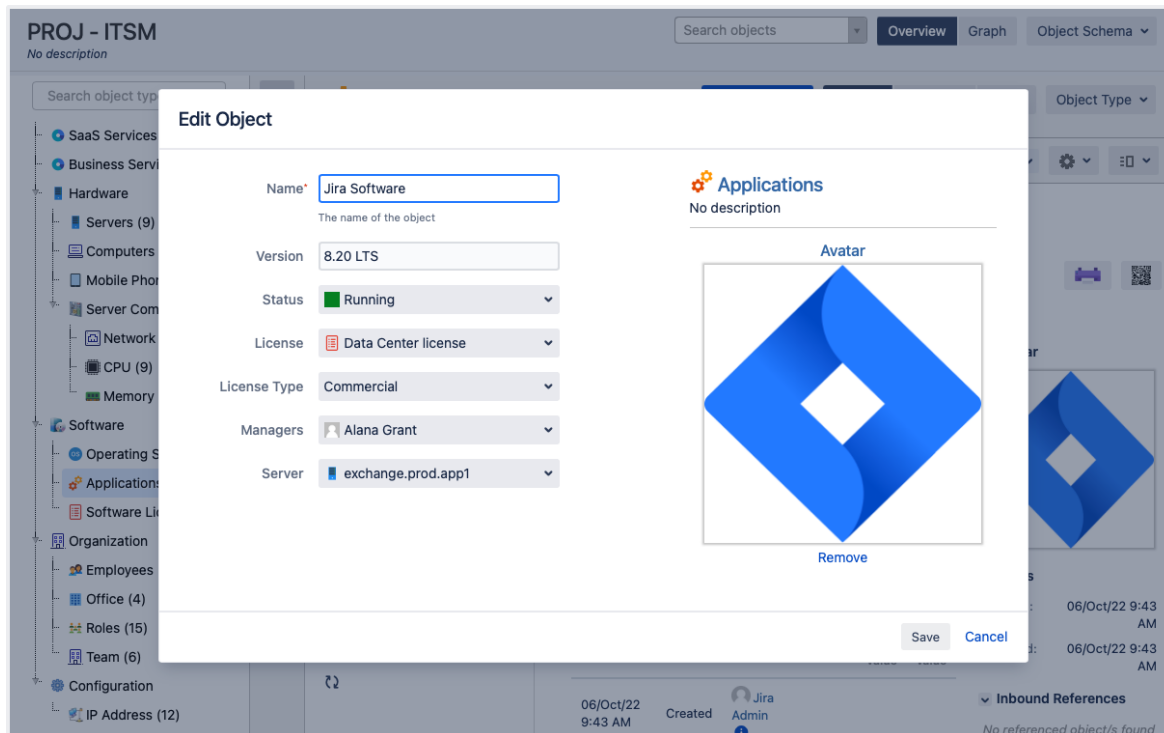
2. Add an attribute of type User or Group. In the following example, we added a User attribute well assign the right user to this object in the next step.



This close-up shows the 'Add' button for a new attribute. The attribute name is 'Managers', the type is 'User', and the 'SHOW ON PROFILE' checkbox is checked. Below the name and type fields, there are two empty input fields, a 'Default' dropdown menu, and a 'Text' dropdown menu. An 'Add' button is located at the bottom right of the form.

3. You can configure extra details for each of your attributes by selecting the cog icon on the right. One of the useful options is **cardinality** it will let you select how many users can be selected for a field.
4. Switch to the **Objects** tab, edit an object, and assign a user or group to the attribute you added.

i If you **select** groups, keep them rather small as bigger groups will affect performance we need to search through every user in the group.



An Assets object field works like a link between Assets and your projects in Jira. The field is configured differently than Jira custom fields, but when it comes to adding it to Jira screens, it behaves like a regular custom field. You will select this field, together with an attribute, when creating an approval step. For more info on the Assets objects field, see [Default Assets custom field](#).

To create the Assets object field:

1. In Jira, go to **Administration > Custom fields**, and create a new custom field.
2. Select the type as *Assets objects*. The object fields marked as deprecated are also supported.
3. Select the name, applicable projects and issue types.
4. Associate the custom field with the screen that are used by the workflow you're planning to add the approval to. Typically, these would be the *Create* and *Edit* screens, related to the issue type used by your workflow.

To link the Assets object field to your projects and objects:

1. On the Custom field page, find the custom field, and select the more options menu > **Configure**.
2. Select **Edit Assets configuration**. You'll now see configuration specific to this field.

3. Select the object schema in which you store your assets. This option should be enough to use the field for the approvals. If youd like to configure more, see the docs linked above.

Configure Assets Custom Field - Affected object (customfield_10203)

Scope

Object Schema	Resources	(i)
Filter Scope (AQL)	Not configured	(i)
Filter Issue Scope (AQL)	Not configured	(i)
Filter Assign Scope (AQL)	Not configured	(i)
Make Filter Assign Strict	<input type="checkbox"/>	(i)

User Interaction

Type of interaction	<input checked="" type="radio"/> Select <input type="radio"/> Checkbox / Radio button	(i)
Filter objects with attributes	Label x	(i)
Object attributes on Issue view		(i)
Multiple	<input type="checkbox"/>	(i)
Option to add all objects	<input type="checkbox"/>	(i)

Jira Service Management

Show object picker on customer portals	<input checked="" type="checkbox"/>	(i)
Force to show on Customer Issue details	<input type="checkbox"/>	(i)

[Return to Jira Custom Field](#)

Adding, configuring, and viewing your approvers

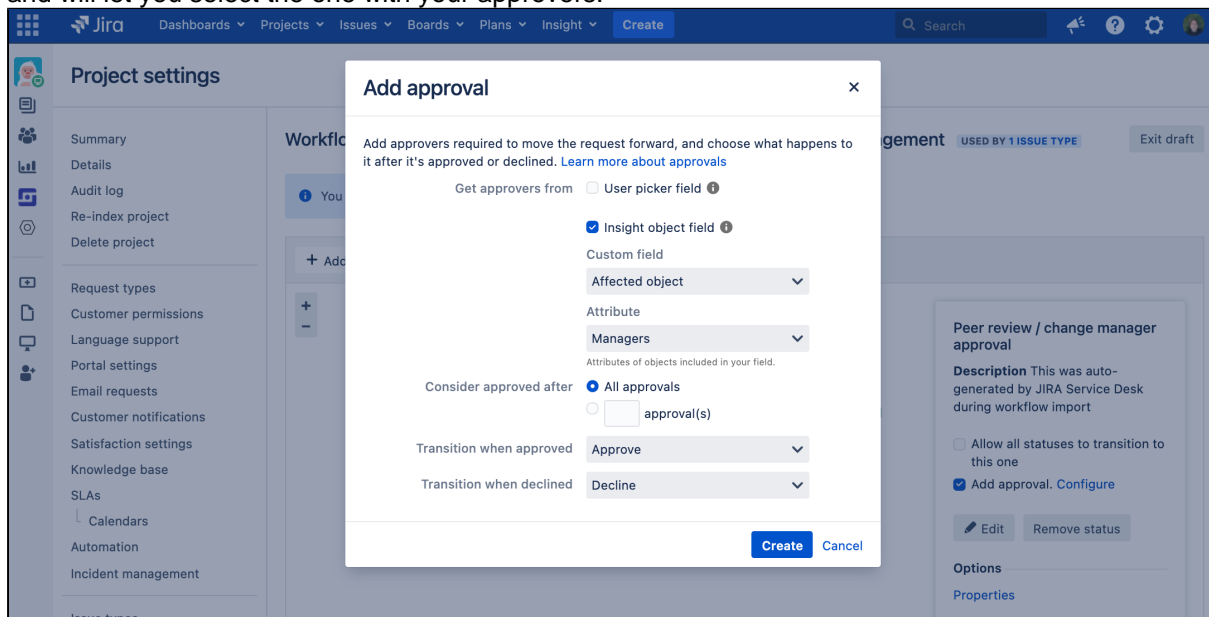
With the attributes and the custom field ready, you can now configure the approval step and add approvers from Assets to your request.

1. Add the approval step to your workflow

To add the approval step to your workflow:

1. Edit the workflow to which you want to add the approval.
2. Select a status, and select **Add approval**.

3. Select **Assets object field** and select your custom field. We'll check this field for any valid attributes and will let you select the one with your approvers.



4. Select how many approvals are required and which transition your request should use when its approved or declined.
5. Select **Create** to add the approval.

2. Enable the Assets object/s field on the customer portal

Approvers change based on the object selected in your request, but someone needs to actually select an object. In this step, you'll let customers select objects on the customer portal. If you skip it, your agents can still select objects in the agent view.

To enable the custom field on the customer portal:

1. In your project, go to **Project settings > Request types**.
2. Select the request type.
3. If your field isn't in **Visible fields** already, select **Add a field**, and add it.

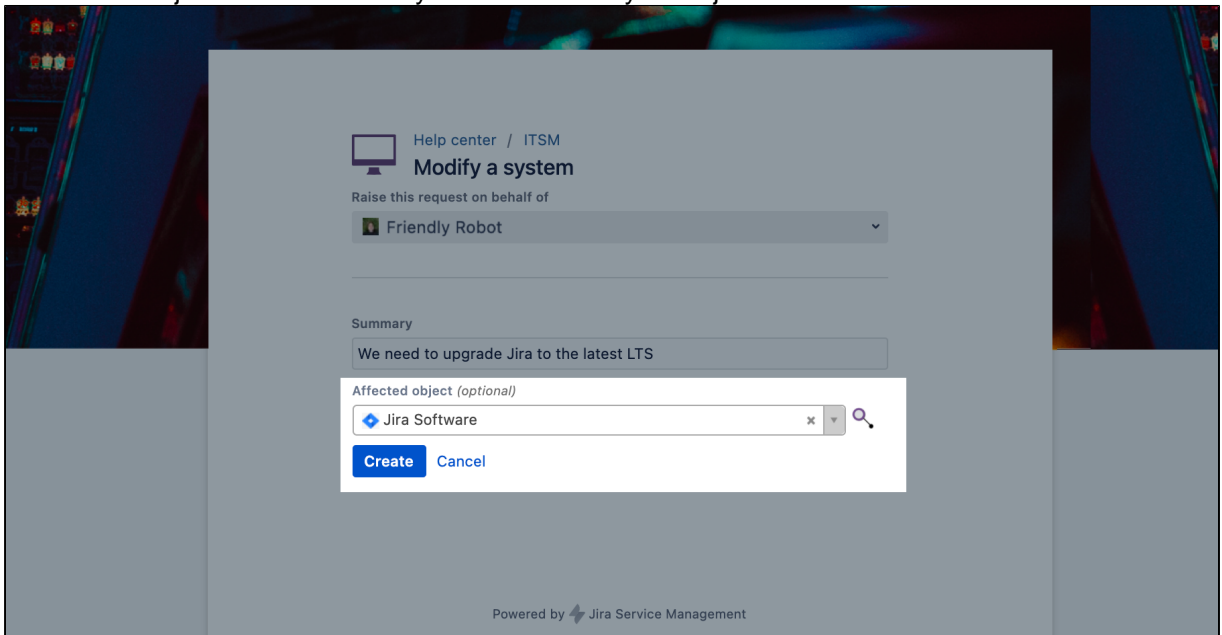
3. Raise a request and view approvers

When you raise a request and select an object in it, approvers will be added automatically when the request reaches your approval step.

To raise a request:

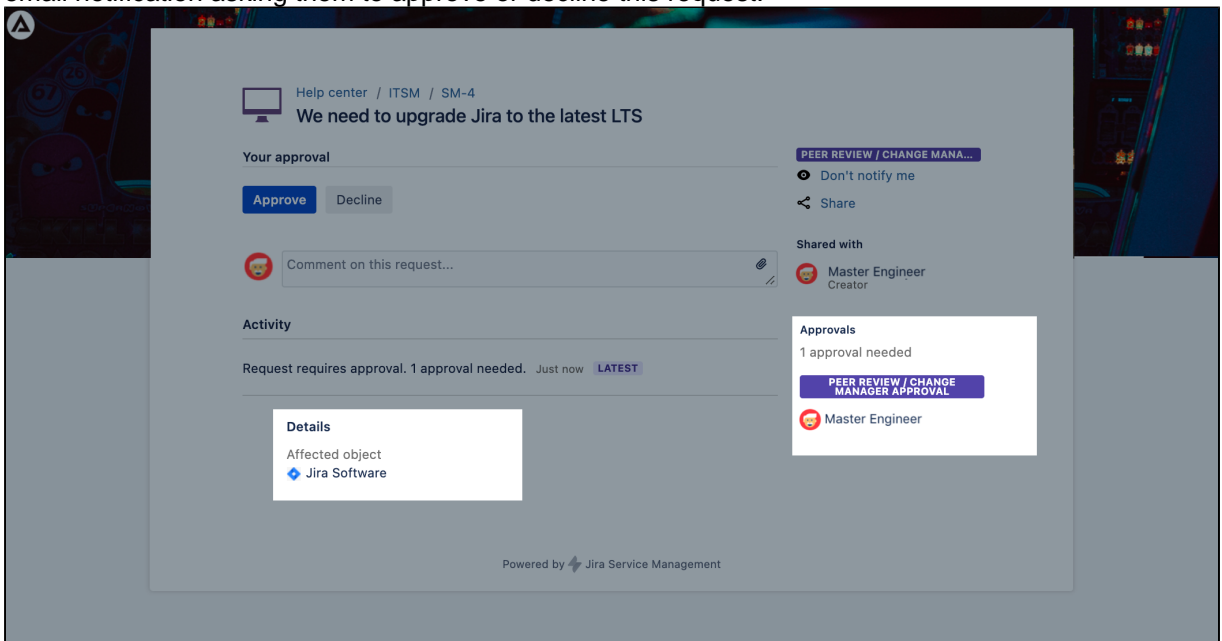
1. Open your customer portal and raise a request.

2. The Assets objects field should let you select one of your objects from Assets.



To view approvers for the request:

1. Transition the request to the workflow status with your approval step.
2. Approvers related to the selected object should be added automatically. They will also receive an email notification asking them to approve or decline this request.



Importing and exporting object schemas

Importing and exporting schemas can be useful if you're using other instances of Assets and want to move schemas between them, or if you exported a schema in the past, and now's the time to bring it back. You can choose to export only schema structure (including object types and their attributes) or also include objects.

Importing an object schema

To import an existing Assets object schema:

1. In the top navigation bar, select **Assets > Configure**.
2. Switch to the **Import/Export** tab.
3. Fill in the schema details. You can find descriptions of available fields in the table below.
4. Once you're ready, select **Import**.

After importing an object schema, you'll need to reindex Assets (not Jira). You can do it by going to **Administration > Manage apps > Indexing Assets**.

References | Statuses | Icons | Roles | Index | Import / Export

i You can import a new object schema from the server.

File name*
The file will be imported from: `/var/atlassian/application-data/jira/import/insight`

Name*
Max. 30 characters.

Key*
Max. 10 characters.

Description
Max. 80 characters.

Import objects Yes No

Include Attachments Object Avatar

Password Show password

Field	Description	Required
-------	-------------	----------

File name	List of previously exported object schemas that are available for import. This includes all schemas that are in the import directory specified below this field. If your schema is not on the list, copy it to this directory.	Yes
Name	Name of the object schema that will be created by the import. The name must be unique.	Yes
Key	Key of the object schema that will be created by the import. The key must be unique.	Yes
Description	Description of your schema.	No
Import objects	Choose whether you'd like to import objects that belong to the schema. Here are some details: <ul style="list-style-type: none"> ▪ Yes (default): Objects will be imported. ▪ No: Only object schema structure will be imported. This includes object types, object type attributes, etc., but not objects themselves. 	Yes
Include	Choose whether to include attachments and avatars of objects.	No
Password	If the file chosen for the import is protected by a password, then that password must be provided in order to perform the import.	No

Exporting an object schema

To export an object schema:

1. In the top navigation bar, select **Assets > Configure**.
2. Switch to the **Import/Export** tab.
3. Fill in the schema details. You can find descriptions of available fields in the table below.
4. Select **Export**.

References
Statuses
Icons
Roles
Index
Import / Export

i You can export an existing object schema to a file on the server.

File name *

Enter a filename (.zip) to export object schema to.
The file will be stored to: /var/atlassian/application-data/jira/export/insight

Object Schema *

Object schema with objects
 Object schema without objects

Password
 Show password

Field	Description
File name	File name for your exported schema. The file will be saved in the directory shown below this field.
Object schema	List of object schemas that can be exported.
Object schema with / without objects	Choose whether to export objects together with the schema, or only the schema structure (object types, object type attributes).
Password	If provided, the exported file will be protected by the password entered by the user. Later when importing this file, the same password must be provided in order to be able to perform the import.

Assets REST API documentation

You can use Assets API to integrate and customize Assets to your needs. You can integrate with an external system to get the data about your assets or create your own post functions.

For Assets 8.10.13 or later and Jira Service Management 5.3 or later, see [Assets REST API documentation](#).

For Insight 8.10.11 or earlier and Jira Service Management 5.2 or earlier, see [Insight REST API documentation](#).

For other advanced information, also check [Advanced usage guides](#).

Advanced usage guides

If you're looking for advanced knowledge, including Assets app development, REST APIs and Java docs, see [Advanced usage guides](#).

Upgrading Insight with Jira Service Management

Insight - Asset Management is part of Jira Service Management Data Center, but its also available as a [separate app on Atlassian Marketplace](#) so people not ready to upgrade their instances can also get it. This page will help you understand which version of the Marketplace app corresponds to Jira Service Management, and will guide you through some upgrade steps if youre using this app.

Versions

The built-in functionality and the Marketplace app are essentially the same. Heres how the apps version corresponds to Jira Service Management:

Jira Service Management	Insight - equivalent Marketplace app
4.18	8.8.0
4.17	8.7.9
4.16	8.7.4
4.15	8.6.13

There were some differences in issues resolved and release timeline in versions 4.15-4.17. Since version 4.18, the built-in functionality and the Marketplace app are aligned.

Upgrading Insight with Jira Service Management

Whether you need to complete any pre-upgrade steps before upgrading Insight or Jira Service Management depends on the Insight version youre using, so lets start with checking that.

Step 1: Check your Insight version

To check which Insight version youre using:

1. Go to **Administration > Manage apps > Manage apps**.
2. Find Insight and expand it to see the version.

Insight - Asset Management is now part of Jira Service Management Data Center.

If you use Jira Service Management Server and already have a license for this app, you can continue to renew your app license. If you don't already have a license you'll need to migrate to Jira Service Management cloud or Data Center to use the app.

You can learn how to configure, use, and share this app by getting started below.

[Get started](#)

No screenshots available	Version: 9.0.11	222 of 227 modules enabled
	Vendor: Atlassian	
	App key: com.riadalabs.jira.plugins.insight	

i If youre seeing Insight in versions 9.x, like on the image above, it means that youre using the built-in Insight functionality in Jira Service Management. You dont have to take any actions and can safely upgrade.

Step 2: Complete the pre-upgrade steps

Look at the versions below and complete the steps required for your version.

Insight 8.4 or later

You're good to go. You're on one of the latest versions and there are no breaking changes.

Insight 8.x - 8.3

Complete these steps before upgrading: [Preparing for Insight 8.4](#)

Insight 7.x

1. Upgrade to the latest version of Insight 7.x, which is 7.1.4.
2. Upgrade to Insight 8.3.
3. Complete these steps: [Preparing for Insight 8.4](#)
4. Upgrade to the latest version of Insight or Jira Service Management, as described below.

Step 3: Upgrade Insight or Jira Service Management

If you completed all the required steps, your app is ready to be upgraded.

- If you're looking to just upgrade the Marketplace app to the latest version, upgrade it in **Administration > Manage apps > Manage apps**.
- If you're upgrading Jira Service Management that has the Insight functionality built in, your app will be automatically upgraded.

How to upgrade Insight to specific versions instead of the latest one

When you try to update the app in **Manage apps**, it will be updated to the latest version, which isn't a good idea if you need to upgrade through intermediary versions first.

To download and install a specific version:

1. Check the [Version history](#) tab on Marketplace.
2. Download your version.
3. In Jira, in **Manage apps**, click **Upload app**. Your version will be uploaded and installed.

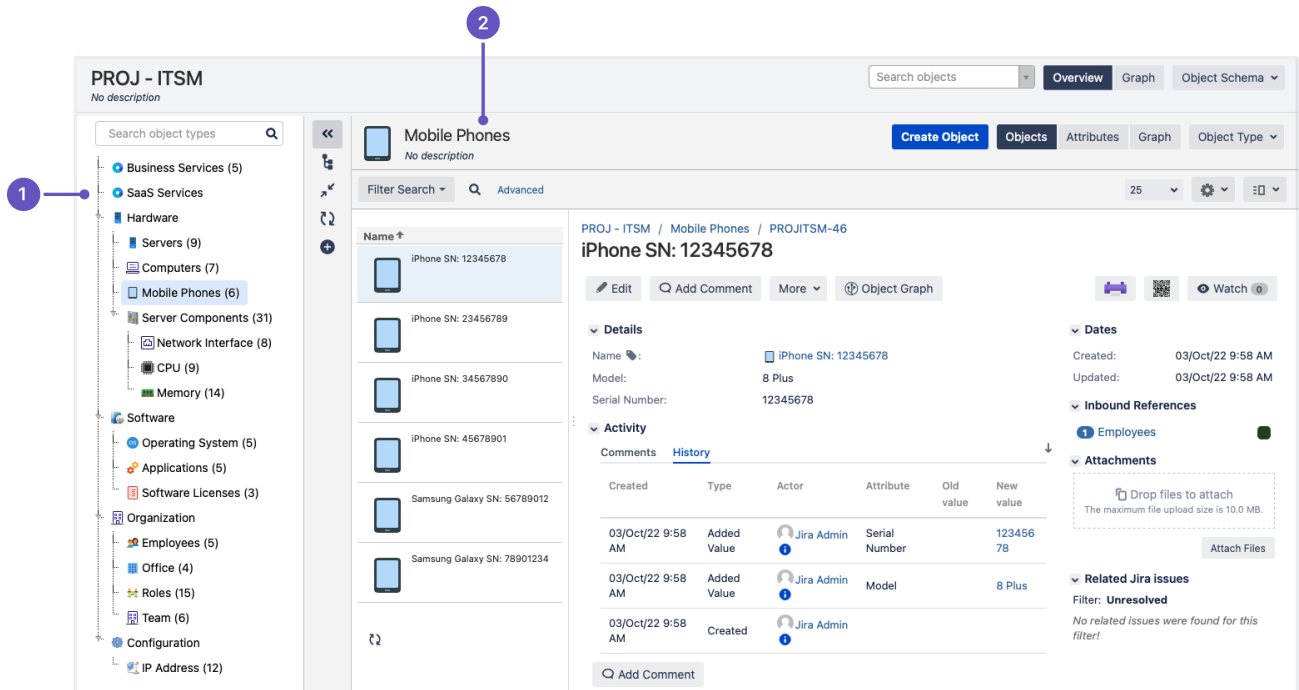
Working with object schemas

An object schema is your actual CMDB. It's a collection of object types and their attributes, objects and references between them. It also includes your configuration items, such as roles, icons, statuses, and automation rules. Object schemas work like maps that hold all of it together. You can have many object schemas, and refer to objects inside them from your issues and requests.

Have one, or many

For example, you might have an HR object schema that captures all relevant HR data that adds value to your HR projects. It can contain information about employees, workplaces, and business areas, as well as information about employee assets, such as keyboards, monitors, and laptops. These can be linked together, so when you click on an employee, you immediately see other objects with information about their office, personally-assigned laptop, or even keyboard.

Here's what a sample schema from the ITSM template looks like:



1. **Object types:** The list of all object types within a schema, organized in a tree structure.
2. **Objects:** The list of objects within the selected object type, and details about them. As you can see, although all these objects are different, they use the same information, like *Name*, *Model*, or *Serial Number*.

Quick overview

Here's a video on object schemas if you'd like to get a quick overview:

Get started with object schemas

Here's a list of pages that have all the info you need:

- [Creating object schemas](#)
- [Configuring object schemas](#)
- [Configuring Assets automation rules](#)
- [Configuring print labels](#)

Creating object schemas

An object schema is a map on which you add various object types and then underlying objects. You need to have an object schema before you can create any object types and objects. [Learn more about object schemas](#)

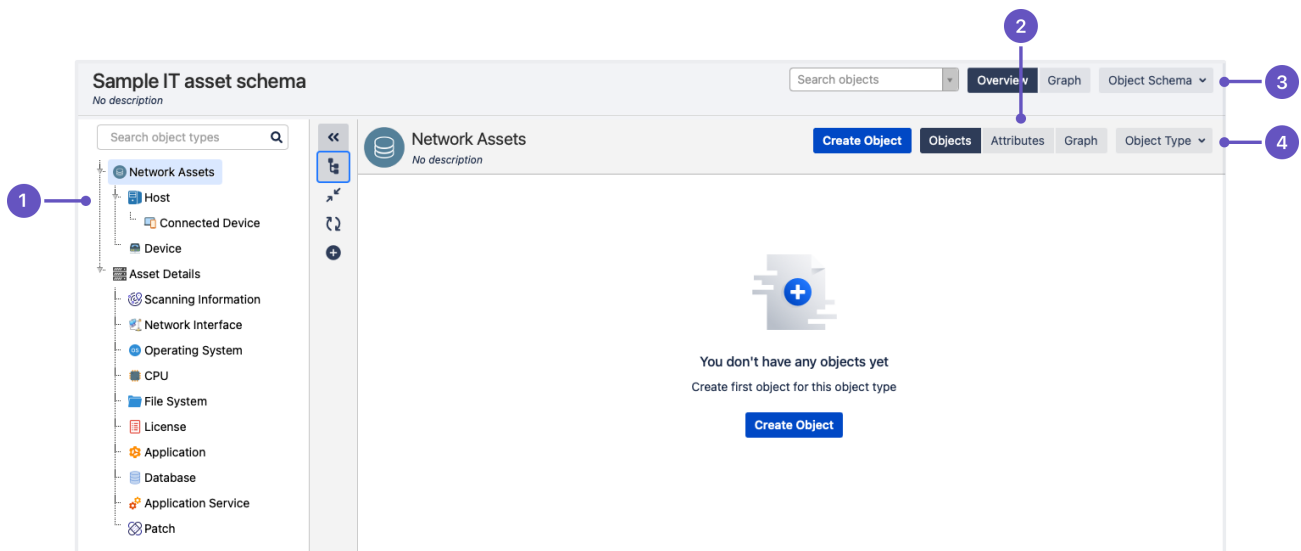
Create an object schema

To create a new object schema:

1. In the top navigation bar, select **Assets > Assets object schemas**.
2. Select **Create object schema**.
3. Select **Empty object schema**. You can also select one of the [schema templates](#) that already come with a specific structure, including object types and attributes.
4. Give it a name, key, and description.
5. Set permissions for your schema, so your users can create object types and objects (see details below).

Sample IT assets schema

Here's the view of an object schema, based on the sample IT assets schema, pre-filled with data.



1. **Object types:**List of object types in a schema.
2. **Object type view:**Switch between viewing object type's objects, attributes, or graph the visual representation of its references.
3. **Object schema actions:**Configuration options for the current object schema.
4. **Object schema actions:**Configuration options for the selected object type.

Set permissions for your schema

Permissions let you choose roles that can view or edit your schema.

i Important

If you want your users to later work on object types and objects, they need to have at least the `Userrole` for the schema itself. Without this role, they won't be able to view or do anything with your schema.

To set schema permissions:

1. Open your schema.
2. At the top-right, select the **Object Schema** dropdown menu, and select **Configure**.

3. Switch to the Roles tab, and add users or groups to specific roles. For more info about permissions, see [Configuring roles and permissions](#).

Configure your schema

Object schemas have several configuration options that let you tailor it to your needs and make the future work easier for your users. For available options, see [Configuring object schemas](#).

Object schema templates

Assets comes with default templates for object schemas. They have some basic structure for the most common uses cases, so you can try them out for some inspiration.

Object schema template come with:

- Object type structure
- Attributes for object types
- References between object types

Sample IT asset schema

Here's the default structure for the IT asset schema:

The screenshot shows the 'Sample IT asset schema' interface. On the left is a navigation tree with categories like Network Assets, Asset Details, and various hardware/software types. The main panel is titled 'Scanning Information' and shows a table of attributes for this object type.

Id	Name	Description	Type	Type Value	Additional Value
503	Key		Default	Text	
505	Created		Default	DateTime	
506	Updated		Default	DateTime	
587	Scan Time		Default	Text	
588	Network Interface		Object		<input type="checkbox"/> Network Interface
589	Used Credential ID		Default	Text	
590	Discovery Version		Default	Text	
591	Scanning Time Elapsed		Default	Float	
592	Pre-Fix		Default	Text	
593	Object Hash		Default	Text	

Sample HR asset schema

Here's the default structure for the HR asset schema:

The screenshot shows the 'Sample HR schema' interface. The main panel is titled 'Organisation' and shows a table of attributes for this object type.

Id	Name	Description	Type	Type Value	Additional Value
659	Key		Default	Text	
660	Name	The name of the object	Default	Text	
661	Created		Default	DateTime	
662	Updated		Default	DateTime	

Sample CRM asset schema

Here's the default structure for the CRM asset schema:

Sample CRM schema Search objects Overview Graph Object Schema

No description

Search object types

- Customer
- Project
- Contract
- Contact
- Type of Contract

Customer

No description

Create Object

Objects Attributes Graph Object Type

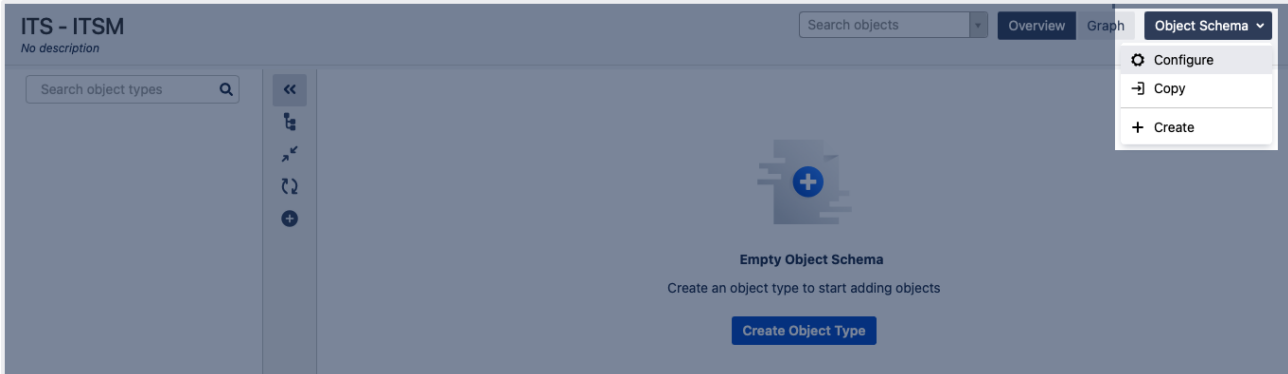
Id	Name	Description	Type	Type Value	Additional Value
734	Key		Default	Text	
735	Name	The name of the object	Default	Text	⚙️
736	Created		Default	DateTime	
737	Updated		Default	DateTime	
754	Country		Default	Text	⚙️
755	Billing Address		Default	Textarea	⚙️
756	Visiting Address		Default	Textarea	⚙️
<input style="width: 100%;" type="text"/>			Default <input type="text"/>	Text <input type="text"/>	Add

Configuring object schemas

You can configure your object schema to change permissions, roles, statuses, references, and more.

To configure your object schema:

1. Open your schema.
2. At the top-right, select the **Object Schema** dropdown menu, and select **Configure**.



Available configuration

Here's what you can do with your schema.

Tab	Description
General	<p>In the General tab, you can change the schema name and description. You can also customize the following options:</p> <ul style="list-style-type: none">Shows a 'create object' icon next to the Assets custom field displayed on issues to allow users to create objects directly from issues (if they can't find their objects). Only available for users with at least the Assets developer role.Enable the possibility to create referenced objects when editing or creating new objects. Only available for users with at least the Assets developer role.If the previous option is enabled, then you can restrict the quick creation of referenced objects to validate all object attributes to respect data consistency.You can choose to publish events related to objects on the Assets activity stream.This option allows other schemas to select objects from this schema. You can select objects only from schemas that have this option enabled.
References	<p>In the References tab, you can configure references that can be later used when linking objects to one another. When users choose to create a reference in their object, they'll see references that you specify here.</p>
Statuses	<p>In the Statuses tab, you can configure statuses that are displayed on objects. For example, you can have statuses like Running and Problem, and choose to change from one to another when an issue is reported against one of your objects.</p>
Icons	<p>In the Icons tab, you can check available icons to be used for object types and objects, and upload new icons.</p>
Roles	<p>In the Roles tab, you can check permissions for you scheme, and add users and groups to specific roles.</p>

Import	In the Import tab, you can import data into your object schema. This is useful if you'd like to import a CSV files with your assets instead of creating objects manually. For more info, see Importing data into an object schema .
Automation	In the Automation tab, you can add automation rules for your schema. For more info, see Configuring Assets automation rules .
Label templates	Templates for print labels. For more info, see Configuring print labels .
Files	In the Files tab, you can view all files added to your object schema and check their usage. The list will show all files added to the schema itself or any object referenced in it. Those can be user avatars or a CSV file that you imported to your schema.

Related pages

Have a look at these pages:

- [Importing your data](#)
- [Working with object types](#)

Configuring Assets automation rules

You can create automation rules that automatically perform actions (for example, notify users) based on specific events (for example, object updated) for all objects, or only a group of objects in a schema.

Before you begin

Make sure you have the right permissions:

- To configure automation rules, you need to have at least the Assets manager role for the schema.
- To run and execute automation rules, the user (either the logged-in user or the user that you select for running scheduled events) needs to have at least the Assets users role for the schema. If this user has a different role for the object types affected by the rule (object type's permissions take precedence over schema's permissions), then also grant them the Assets users role for these object types. This is only needed if a rule affects such object types, otherwise permissions are inherited from the schema.

For more info about permissions, see [Configuring roles and permissions](#).

Create a rule

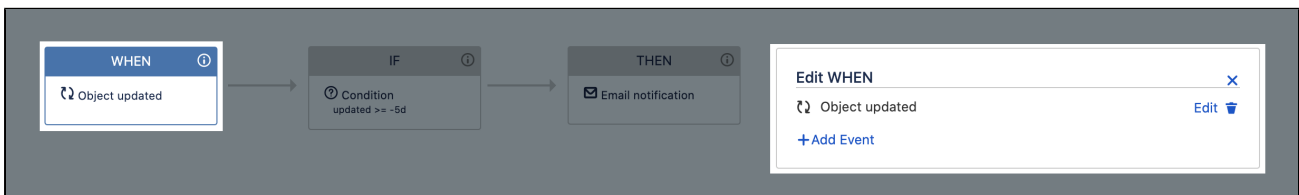
1. Open your schema, and select **Object schema > Configure**.
2. Switch to the **Automation** tab.
3. Create a rule.
4. Enter the rule's name and description.
5. Select each of the boxes (WHEN, IF, THEN). A panel will appear on the right allowing you to add an event, condition, and action. You can read more about them below.

i Any changes to your rule will only be saved after you select **Save**. If you've deleted something by mistake, you can select **Cancel** to revert these changes.

Configure your rule

Configure your rule by specifying events (WHEN), conditions (IF), and actions (THEN).

1. Add events (WHEN)



1. Select the WHEN box.
2. On the right, select **Add events**.
3. Select the desired event, then select **Add**.

✓ Good to know

- You can add up to five events for a single rule.
- Instead of creating two or more rules with the same actions (for example, email notification), add two or more events to the same rule. It will be easier to maintain later.

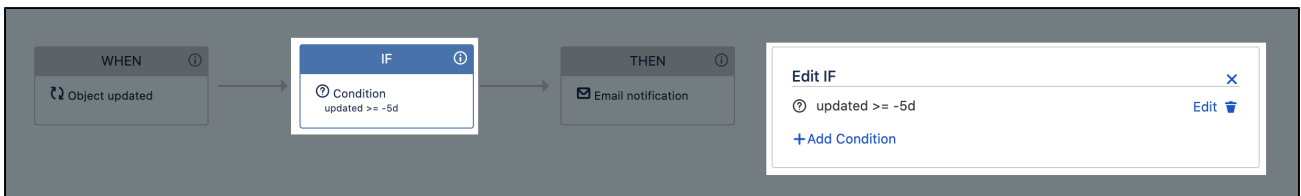
Event types

Event type	Description
------------	-------------

User events	Most events, except the Scheduled event are triggered when a logged-in user performs an action on an object (object updated, attachment added, etc.) These events don't require any additional configuration.
Scheduled events	<p>Scheduled events aren't triggered by an action performed by a user. They run according to the specified cron expression (e.g. once a day, once an hour) and conditions.</p> <p>Here are the required parameters for scheduled events:</p> <ol style="list-style-type: none"> Condition: This is an IQL (Assets Query Language) expression. When the scheduled rule is activated, all objects that match the condition will trigger individual actions. Cron expression: This determines when and how often the provided condition will try to match Assets objects. By default, the expression is set to "0 0 12 1/1 * ? *", that is: once a day at noon. Learn more about cron expressions Triggered rule will run as: You need to select a specific user here because scheduled events can't be triggered by a logged-in user. You can select the user on the top-right of the page.

2. Add conditions (IF)

You can add a condition if you don't want the configured action to apply to all objects in your schema, but rather a specific set of objects.



1. Select the IF box.
2. On the right, select **Add condition**.
3. Enter an AQL statement and select **Add**.

Good to know

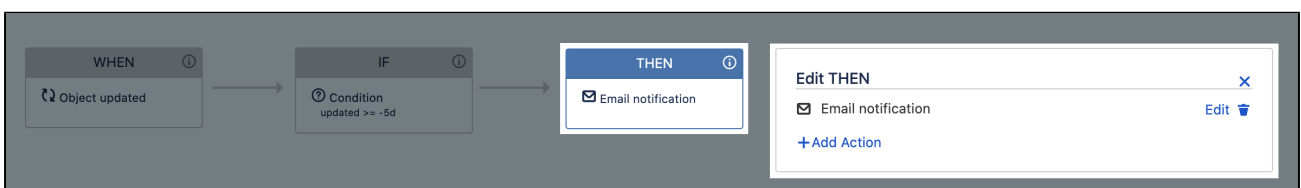
- A condition (IF) always needs a THEN box. When you add another condition, you'll be able to edit its actions.
- You can configure up to three IF/THEN pairs.

Example

- A rule is triggered by an object updated event.
- The rule has a condition: "name = Windows server".
- The rule has an email notification action.


The rule configuration in this example means that if a user updates an object whose exact name is Windows server, an email notification is automatically sent.

3. Add actions (THEN)



1. Select the THEN box.
2. On the right, select **Add action**.

3. Select your action and fill in the details. Every action requires different information. For example, details needed for Create a Jira issue will differ from those needed to Execute a Groovy script.
4. Select **Add**.

 **Good to know**

- You can have up to five actions for every THEN box.

Available actions

Here's a list of actions that you can select:



This action allows the rule to change the value of an object's attribute.

Set an attribute value of the Assets object affected by the event. Multiple values are separated by a comma (,).

Field	Single value example	Multiple values example
Attribute name	Status	Server
Value	Stopped	Server 1,Server2

This action allows the rule to create a Jira issue.

 **Good to know**

-  The issue type selected must be available for the chosen project, otherwise a Jira issue can't be created.
-  If the custom field is populated, the chosen project must be configured to handle this object type, otherwise this field won't have any effect.

The screenshot shows a configuration form for the 'Create JIRA Issue' action. The fields are as follows:

- Action:** Create JIRA Issue
- Project:** DevOps
- Issue type:** Task
- Priority:** Highest
- Assignee:** Automatic
- Custom field:** Maintenance Host
- Summary:** Server `${label}` stopped!
- Description:** Server details - name: `${label}`, key: `${Key}`, id: `${id}`
- Idle interval:** (empty field) ⓘ

Below the 'Idle interval' field, there is a note: "The wait time before the action can execute again, once executed."

Fields that support placeholders	Syntax example	Result example
Summary	Server <code>\${label}</code> stopped!	Server <code>\${exchange.prod.main1}</code> stopped!
Description	Server details - name: <code>\${label}</code> , key: <code>\${Key}</code> , id: <code>\${objectId}</code>	Server details - name:exchange.prod.main1, key:WS-5969, id: 17400

This action allows the rule to send email notifications related to the event that was triggered.

Example: For the object `updatedevent`, an email will be sent showing the changes related to this object.

Note that for the **Recipients by attribute** option you need to select the object type in the rule's condition with attributes of the email, user, or group type.

Action *

JIRA Recipients *

External Recipients

A comma separated list of emails (a@c.com,b@c.com,...).

Recipients by Attribute

Attribute(s) of type: Email, User or Group

Subject *

Message *

Idle interval

The wait time before the action can execute again, once executed.

Fields that support placeholders	Example syntax	Example result
Subject	Server: \${label} is stopped.	Server:exchange.prod.main1is stopped.
Message	Server: \${label} (\${Key}) has status: \${Status}.	Server:exchange.prod.main1(W5-5969) has status: Stopped.

Groovy scripts need adequate permissions and must be whitelisted to be executed. [Learn how to add Groovy scripts to the Whitelist](#)

This action allows a triggered rule to execute a Groovy script.

Action* ▾

File path*
Absolute file path, i.e: /Users/MyName/somefile.groovy

Idle interval ⓘ
The wait time before the action can execute again, once executed.

Data available to the Groovy script by our rule engine

Parameter	Description	JAVA class
object	The object that was directly involved in a user action (like updating an Assets object) or matched by a scheduled rule through AQL, with an "Execute Groovy Script" as an action object that was directly involved in a user action (for example, updating an object) or matched by a scheduled rule through AQL, with an "Execute Groovy Script" action.	com.riadalabs.Jira.plugins.assets.services.model.ObjectBean
currentUser	The user who triggered the rule or a specific user chosen to run the rule.	com.atlassian.Jira.user.ApplicationUser
log	An instance of the Apache Log interface, for convenient logging in a script.	org.apache.commons.logging.Log
objectUpdateList	A list of object attributes that are affected in an object update event that includes an affected attribute in an object, as well as values added or removed for it.	List<com.riadalabs.jira.plugins.assets.services.model.ObjectUpdateBean>

This action allows a triggered rule to make an HTTP request. It supports the standard HTTP methods: GET / POST / PUT / DELETE.

Action*

Url*

User name

Password

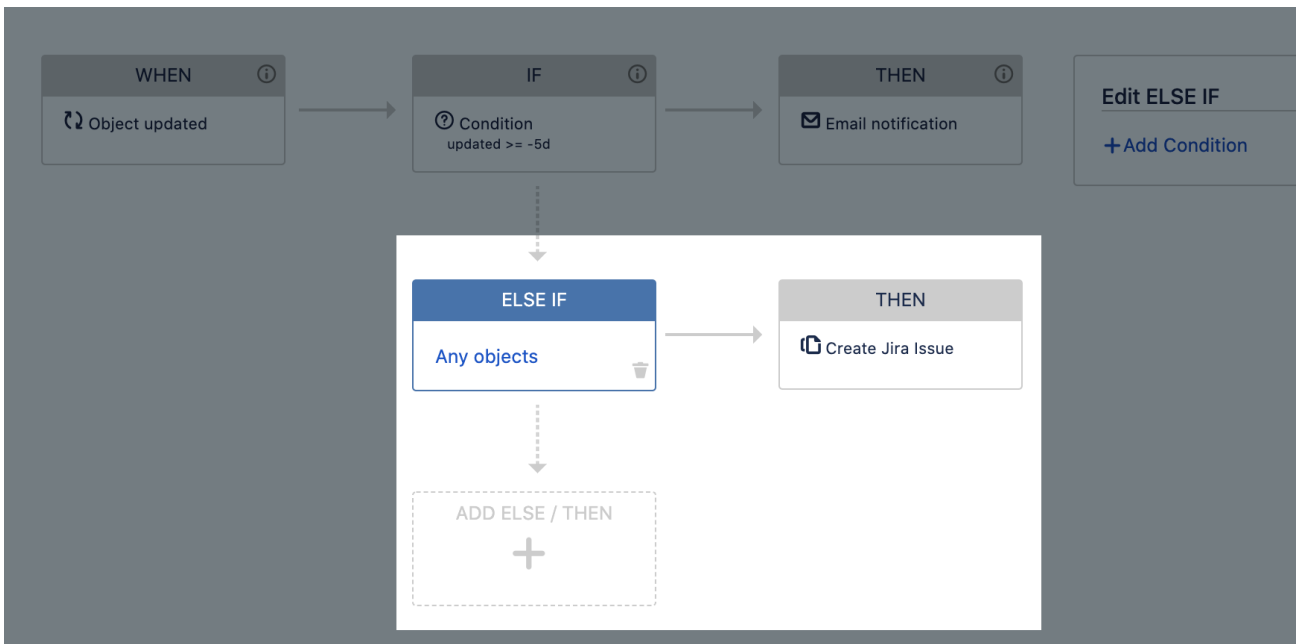
Http method*

Idle interval ⓘ
 The wait time before the action can execute again, once executed.

Fields that support placeholders	Syntax example	Result example
Url	http://company/service?id=\${objectId}&name=\${label}	http://company/service?id=1405&name=exchange.prod.main1
Post Data (POST / PUT)	{ "name": "\${label}", "key": "\${Key}", "id": "\${objectId}" }	{ name:exchange.prod.main1,key:WS-76100,id:76100 }

4. Add more conditions and actions

A rule can only consist of one event (WHEN), but it can have up to three pairs of conditions and actions (IF /THEN). This allows you to add more conditions and actions, if the first ones aren't matched.



1. Select the ELSE IF / THEN box.
2. Add actions, and conditions if needed.
3. Select **Save**.

More automation

Automation rules are related to a specific object scheme, but there's way more automation that you can add. Your Jira admin can add conditions and validators, as well as post-functions, to Jira workflows that will make issues affect objects while they're being transitioned through workflows. For more info, see [Adding Assets functions to Jira workflows](#).

Configuring print labels

You need to create at least one print labels template before you can print labels for your objects. Templates are created for each object scheme separately.

To create a template:

1. Open your object schema.
2. At the top-right, select **Object schema > Configure**.
3. Switch to the **Label templates** tab.

Template configuration

Here's the explanation for fields that you see when creating a new template or editing an existing one.

Responsive height

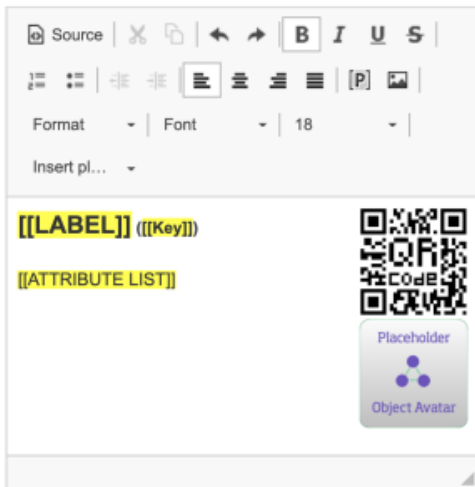
You can select this option to give your template responsive height, allowing it to expand to fit the content.

Border

You can select this option to add a border to your template. Here's an example of a template with and without a border:

Template

This editors allows you to create a customized template. It supports placeholders and images, and comes with some styling options. When you create a new template, the editor will already include a default template.



Source gives you the possibility to edit the html source for the template.

```

Source
<table cellpadding="0" cellspacing="0" style="width: 100%">
  <tbody>
    <tr>
      <td style="max-width: 80%; vertical-align: top;"><strong><span style="font-size:18px;">
[[LABEL]]</span></strong>&nbsp;<strong><span style="font-size:12px;">([[Key]])</span></strong>
      <div style="margin-top: 20px;">
        <p>[[ATTRIBUTE LIST]]</p>
      </div>
    </td>
    <td style="width: 20%; max-width: 100px; vertical-align: top;"></td>
    </tr>
  </tbody>
</table>

```

You can use the **Insert placeholders** drop-down to add any of the attributes available for this object schema. One of the first attributes is "Attribute list" when selected, it will add an additional configuration when printing a template, where you can select the attributes you want.

Some things to watch out for

- Some attributes may be specific to certain object types. Using the same template for other object types will render the text as "No value" on the printed labels.
- Additionally, if you have a language setting enabled for Assets, for example Spanish, the "No value" text will be printed as "Sin valor". Well, at least everybody will get it.

Adding images

You can also add images to your templates. This can be pretty much anything you need in your template from QR codes to logos or icons. We've added some default images for you, so feel free to play around.

Printing labels and QR codes

Once you're ready with the template, you can try printing some labels. For more info, see [Printing labels and QR codes](#).

Working with object types

An object type groups objects that use the same kind of information, conveyed through their common attributes. Rather than a single PC, your object types would be Computers, Hardware, Software, Employees, and so on. You can create as many different object types as you like, and then group your various objects within. Object types can be whatever you want them to be as Assets is very open and flexible.

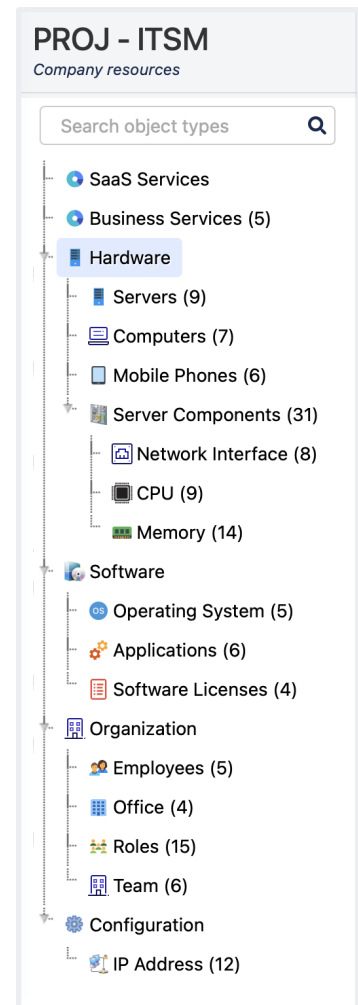
Common object types include:

- Business services
- Hosts
- Laptops
- Software

But they don't have to be IT assets. For example, many people add other useful information, such as:

- Vendors
- Locations
- Employees
- Business priority

You can organize object types in the hierarchy tree in a way that makes sense. This tree is mainly for navigation and readability as you can have empty object types here (e.g. Hardware in the image on the right).



Define your object types with attributes

Attributes is what defines your object types (and underlying objects). They work like fields to which you add data. Jira's issue fields would be the best comparison. You'd choose attributes for your object types, and then the same attributes need to be filled in for underlying objects, whenever someone creates them. You can have as many attributes as you need, and they also come in different shapes and sizes: text, dates, references to other objects, statuses, and so on.

Here's a sample object type with its attributes:

The screenshot displays the 'PROJ - ITSM' interface for configuring 'Servers' object types. On the left, a sidebar lists various object types, with 'Servers (9)' selected. The main area shows a table of attributes for the 'Servers' object type. The table has columns for 'Id', 'Name', 'Description', 'Type', 'Type Value', and 'Additional Value'. The 'Status' attribute (Id 82) is highlighted, showing a dropdown menu with options like 'INCIDENT IN PROGRESS', 'RUNNING', and 'STOPPED'. The right-hand panel shows the configuration for the 'Status' attribute, including a dropdown for 'Type Value' and an 'Add' button.

Id	Name	Description	Type	Type Value	Additional Value
13	Key		Default	Text	
14	Name	The name of the object	Default	Text	
15	Created		Default	DateTime	
16	Updated		Default	DateTime	
82	Status		Status	INCIDENT IN PROGRESS RUNNING STOPPED	
83	Environment		Default	Select	
84	Operating System		Object	Operating System	Operating System
85	IP Address		Object	IP Address	IP Address
86	Server Owner		Object	Employees	Employee
87	Last Serviced		Default	Date	
420	CPU		Object	CPU	A reference

- Object types:** List of object types within a schema, with Server selected.
- Attributes:** List of attributes configured for this specific object type.
- Type:** Type of attributes, here you can see the default ones or object (references to other objects)
- Type and additional value:** Extra values that depend on the attribute type. Note statuses these will change through automation when an incident is raised against one of the underlying servers.

Quick overview

Here's a video on object types if you'd like to get a quick overview:

Get started with object types

Here's a list of pages that have all the info you need:

- [Creating object types](#)
- [Adding attributes to object types](#)
- [Configuring inheritance of attributes](#)
- [Customizing the object view layout](#)

Creating object types

Object types work like containers, or categories. Depending on your structure, they can group other object types or just objects. For example, you can have an object type Hardware that groups child object types Server, Computer, and Mobile phones. Each of these can then group specific objects. [Learn more about object types](#)

Create an object type

To create an object type:

1. Open your object schema.
2. At the top-right, select **Object type** > **Create** (you can also select the plus sign in the vertical menu)
3. Choose the name, icon, and description for your object type.
4. Select the object type's parent. If you select **None**, the object type will be at the top of your asset tree.

Example object type

Here's the view of an object type, based on the sample IT assets schema:

The screenshot shows the 'PROJ - ITSM' interface. On the left, a sidebar lists object types such as SaaS Services, Business Services, Hardware, Servers (9), Computers (7), Mobile Phones (6), Server Components (31), Network Interface (8), CPU (9), Memory (14), Software, Operating System (5), Applications (6), Software Licenses (4), Organization, Employees (5), Office (4), Roles (15), Team (6), and Configuration. The main area displays the 'Servers' object type configuration. At the top right, there are tabs for 'Overview', 'Graph', and 'Object Schema'. Below these are buttons for 'Create Object', 'Objects', 'Attributes', 'Graph', and 'Object Type'. A table lists attributes for the 'Servers' object type, including Key, Name, Created, Updated, Status, Environment, Operating System, IP Address, Server Owner, Last Serviced, and CPU. Each attribute row includes columns for Id, Name, Description, Type, Type Value, and Additional Value. A 'Status' attribute is highlighted with a red 'INCIDENT IN PROGRESS' indicator. At the bottom right, there is an 'Add' button and a link to 'Add parent attributes'.

1. **Object types:**List of object types in a schema.
2. **Object type view:**Switch between viewing object type's objects, attributes, or graph the visual representation of its references.
3. **Object type actions:**Configuration options for the current object type.
4. **Attributes:**List of object type's attributes. These attributes will apply to all child objects.

Next steps

Object types need attributes that define them, so let's add some:

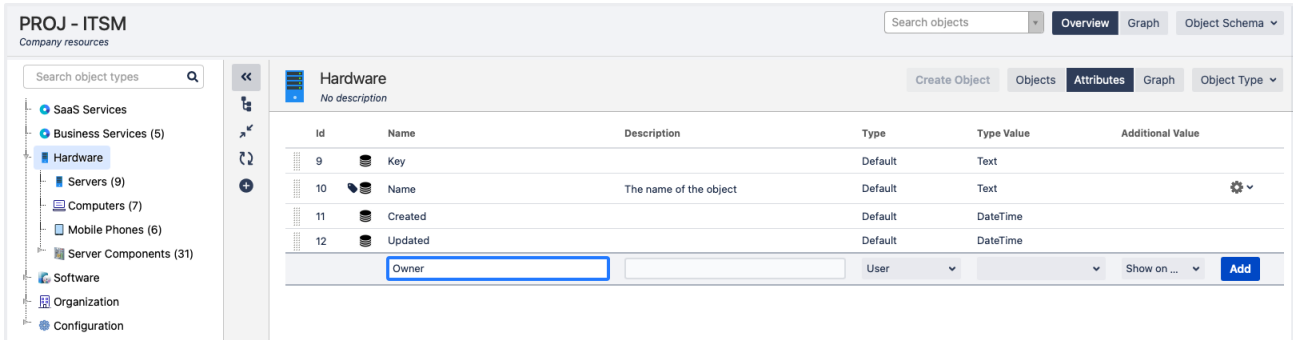
- [Adding attributes to object types](#)

Adding attributes to object types

Attributes are fields that let you add important information to your objects. It can really be anything that you need. You specify attributes for an object type, and then they're passed on to all child objects, requiring users to fill them in. An object type comes with a default set of attributes.

To add attributes to an object:

1. Select your object type.
2. Select **Attributes** to switch to the right view.
3. Add an attribute by entering its name, type, and values. You can read more about them below.



Attribute types

You can select the attribute type, this setting determines how the attribute value should be managed (and if its allowed).

Attribute type	Type value	Additional value	Description
Default	Text	-	Data type with text representation, often used to show normal text. Max 450 characters.
	Boolean	-	Data type with only two possible values: true or false.
	Integer	-	Commonly known as a "whole number", is a number that can be written without a fractional component. For example, 21, 4, and 2048 are integers. This Attribute supports the Long Integer and will allow numbers that range from -2,147,483,648 to +2,147,483,647
	Float	-	Data type representing numbers with decimals.
	Date	-	Data type representing a Date field.
	DateTime	-	Data type representing a Date and Time field.
	URL	-	Data type representing an URL field. May be used for URL Ping service that pings the address every 5 minutes from the server side. Watch object to get email notifications on URL Ping Up /Down.
	Email	-	Data type representing an email field.

	Textarea	-	<p>Data type with text representation, often used when showing large text. Use Assets rich editor to customize the content. Unlimited characters in comparison to the Text attribute.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p>i For developers: Missing attribute beans</p> <p>Here's some info for developers trying to retrieve beans for this attribute using Java API.</p> <p>When you access an Assets object loaded from a custom field via Java API, attribute beans for this attribute type will be missing. You can read more about this bug here.</p> <p>This is because the <code>textarea</code> attribute is not indexed by default. To solve this issue:</p> <ol style="list-style-type: none"> 1. Select Configure for your <code>textarea</code> attribute. 2. In the General tab, select Indexed. </div>
	Select	-	Data type to represent text values, predefined as options.
	IP Address	-	Data type to represent IP Addresses (IPv4)
Object	Assets object	Reference Type	This type enables a reference to another object.
User	Jira group	Show on Profile	This type makes it possible to select a user from the selected Jira group and connect objects with users.
Confluence	Confluence instance	Confluence page	This type enables a link to a Confluence page.
Group	-	Show on Profile	This type makes it possible to select a Jira group and connect object(s) with user(s) in specified groups.
Version	Jira Project	-	This type makes it possible to reference a Jira Version from a specific Jira Project to your object(s).
Project	None	-	This type makes it possible to reference a Jira Project to your object(s).
Status	Allowed Statuses	-	This type is used to set a Status of an object. Define the statuses that should be allowed, and left empty means all statuses allowed.

Attribute configuration

Some attribute types have additional configurations that let you specify default suffixes or options, cardinality, and filtering. These configuration options depend on the type of attribute you've added.

1. To configure an attribute, select the **cog icon** next to it.
2. Adjust the configuration to your needs. Below you can find some details about each configuration option.

General

Option	Description
--------	-------------

Suffix	For default type "Integer" and "Float" you can set a suffix for the attribute. Example is "\$" for an att
Hidden	Hide the attribute from users in the "assets-users" role.
Include children	Include objects from a child object type(s).
Unique	Validate attribute values to be unique within the object type.
Show Sum	For default type "Integer" and "Float" you can choose to add the values and display the sum of the

Indexed

This is a checkbox and you can use this to determine if you want to index this attribute or not.

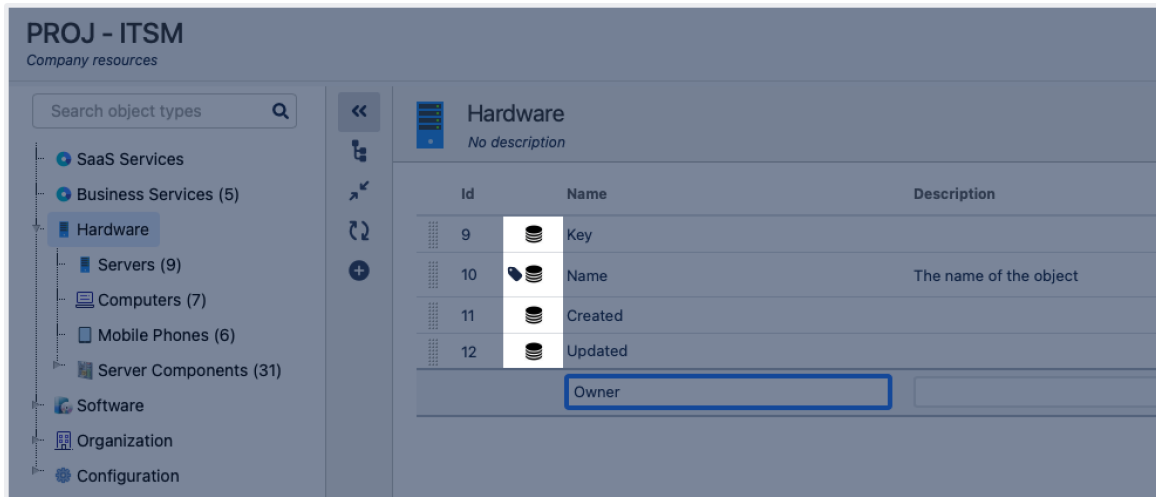
The Indexed feature on every attribute essentially helps you to retrieve your search results faster.

If your attribute is indexed, Assets will not retrieve results from the database every time but instead cache them.

For non-indexed attributes, memory consumption will be lower.

If you index an attribute, a small icon (🗄️) precedes the *Name* of the attribute as shown below. This icon indicates that the attribute is indexed.

Additionally, for the *label* attribute, the indexed icon will appear adjacent to the icon for the *label* (🏷️).



By default, Assets will index all attributes on an object type, but the "Textarea" attribute is an exception. The **Key**, **Label**, **Created**, and **Updated** attributes will always be indexed, as will any attributes of type "Text".

Cardinality

You can specify a minimum and maximum of number of attributes values that can be associated to the attribute. This is common when you need to set an attribute multiple choices or required / mandatory.

For attributes of type "Default" you can only set maximum cardinality to 1, except "URL" and "Email" where you may space separate.

Validation

You may validate attributes of default type "Text", "Email" & "URL" with regular expressions. This can be handy if you want to validate specific information, like an IP address, a domain name, a phone number, or anything else that may require validation.

Options

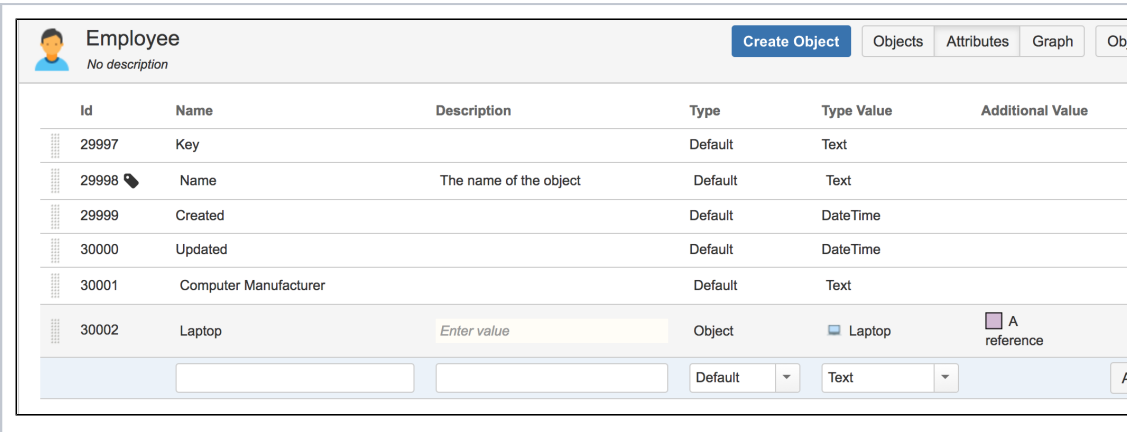
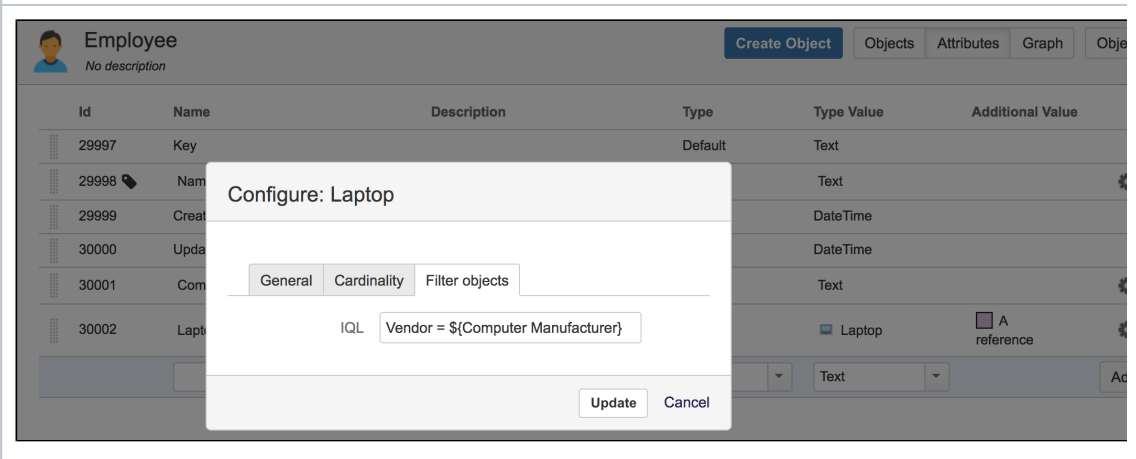
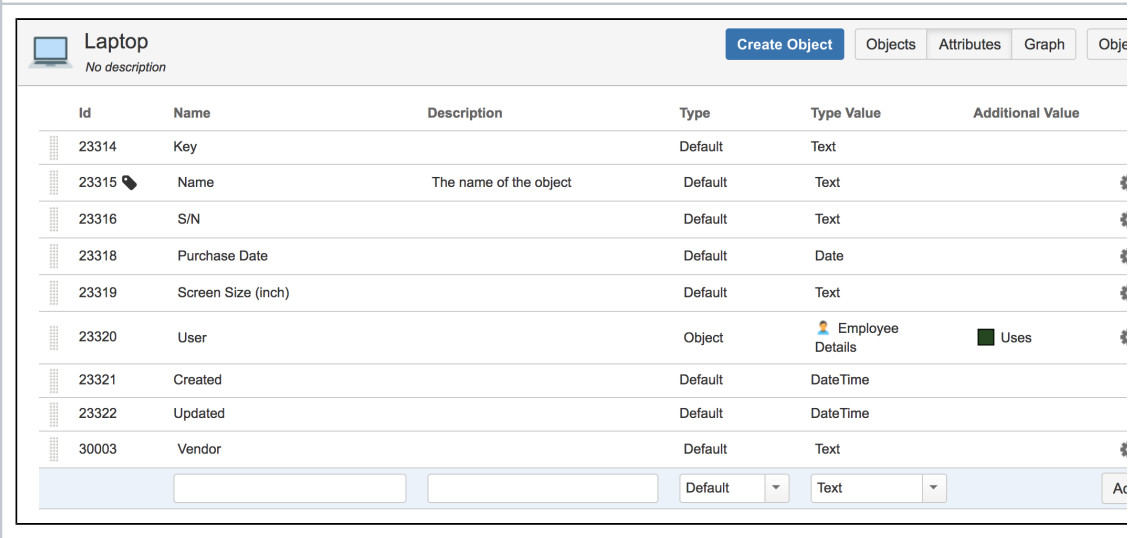
You may add multiple Options to a "Select" Type Attribute by adding them here.

Filter objects

For attribute of type "Object" you may filter objects to be selected by AQL. By using this filter it is possible to create dependencies to other fields when creating/editing objects.

The following examples shows how to configure an object reference attribute on the object type Employee to reference a Laptop. If a value is entered in the Manufacturer textbox only Laptops with the corresponding Vendor will be choosable. It is possible to combine multiple fields in the AQL to get multiple dependencies.

Example description	Example screenshot
----------------------------	---------------------------

<p>The employee has an outbound reference to Laptop. The employee also has an attribute called Computer Manufacturer.</p>	
<p>The AQL for attribute Laptop is configured with: <code>Vendor = \${Computer Manufacturer}</code></p>	
<p>The laptop has an attribute called Vendor.</p>	

Next steps

Have a look at these pages:

- [Configuring inheritance of attributes](#)
- [Customizing the object view layout](#)
- [Working with objects](#)

Configuring inheritance of attributes

You can configure object types to inherit attributes from their parent object types. This is useful if you need to create multiple object types, nested in one another, and want them to keep the same structure.

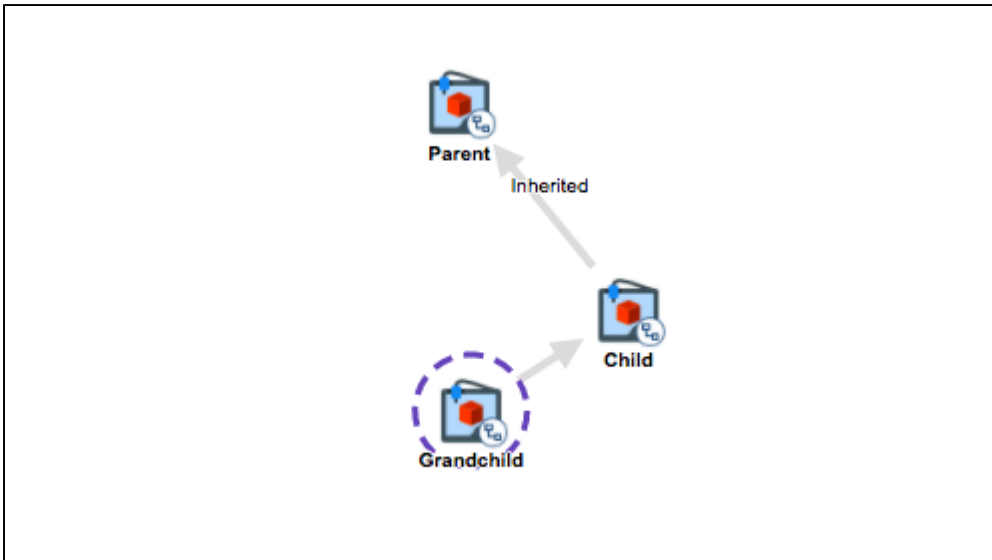
A good example for inheritance is an object type **Host**, with child objects **Linux Host** and **Windows Host**, which inherit common attributes, such as IP Address and Host name.

i You won't be able to set the inheritance for an already existing structure of object types. You can only do it when creating a parent, and then child object types.

To enable inheritance of attributes:

1. Create the first, parent object type.
2. Select **Object type > Configure**.
3. Select the **Let the attributes be inherited by the subclasses** checkbox.
4. Optionally, set this object type to **Abstract** to prevent creating any objects in it. This can be useful if you want the object type to exist only to pass on the attributes, without grouping any objects.
5. Create child object types.

The inheritance will work on the grandchildren and their children and so forth.



Related pages

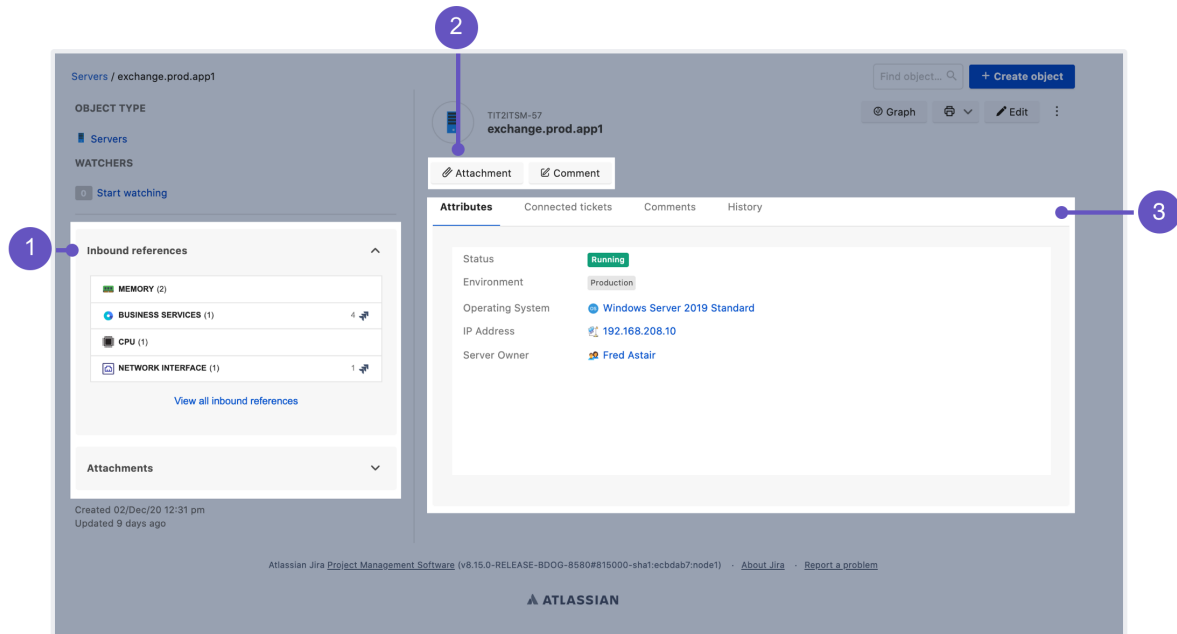
Have a look at these pages:

- [Customizing the object view layout](#)
- [Working with objects](#)

Customizing the object view layout

When you open an object, you'll see some details about it. That's usually some basic information about the object itself, its attributes, connected Jira issues, and so on. This information is presented through widgets you can remove these widgets or add some more to always see the info you're most interested in. You make these changes not on objects specifically, but on object type the changes then apply to all objects within this type.

Here's an example of an object. We described the panels, actions, and widgets that you can see and customize below:



1. **Side panel:** Two widgets are shown on the panel: Inbound references (expanded) and Attachments (collapsed).
2. **Attachments and comments:** These two actions are enabled, allowing users to add attachments and comments to your object.
3. **Main panel:** Several widgets are shown as separate tabs. Users can switch between them to get the info they need.

Customizing the page layout

1. Select the object type for which you want to customize the page layout.
2. In the top-right, select **Object type > Configure**.
3. In the top-right, select **Configure page layout**.
4. Read on for more information on what you can do with specific panels.
5. Once you're done with your changes, select **Save changes**.

Here's a brief description on what you can do in specific panels:

Panel	Description
Side panel	Here you can add widgets that will be displayed below each other. To add a new widget, select Add panel , or just edit the widget that's already there. For the list of available widgets, see Adding widgets .
Attachments and comments	Here you can choose whether users can add attachments and comments to your objects. Clearing any of the check boxes will hide the related buttons from your objects.

Main panel (tabs)	<p>Here you can add widgets that will be displayed in tabs in the main part of your object. To add a new widget, select the plus sign and choose the widget you want. For the list of available widgets, see Adding widgets.</p> <p>Good to know</p> <ul style="list-style-type: none">• Double-click the tab to change its name• Drag & drop tabs to rearrange them
--------------------------	--

Reverting your changes

You can always reset the page layout to the default one by selecting **Reset**. This option will appear only after you save your changes.

Related pages

Have a look at these pages:

- [Working with objects](#)
- [Viewing objects in the object view](#)

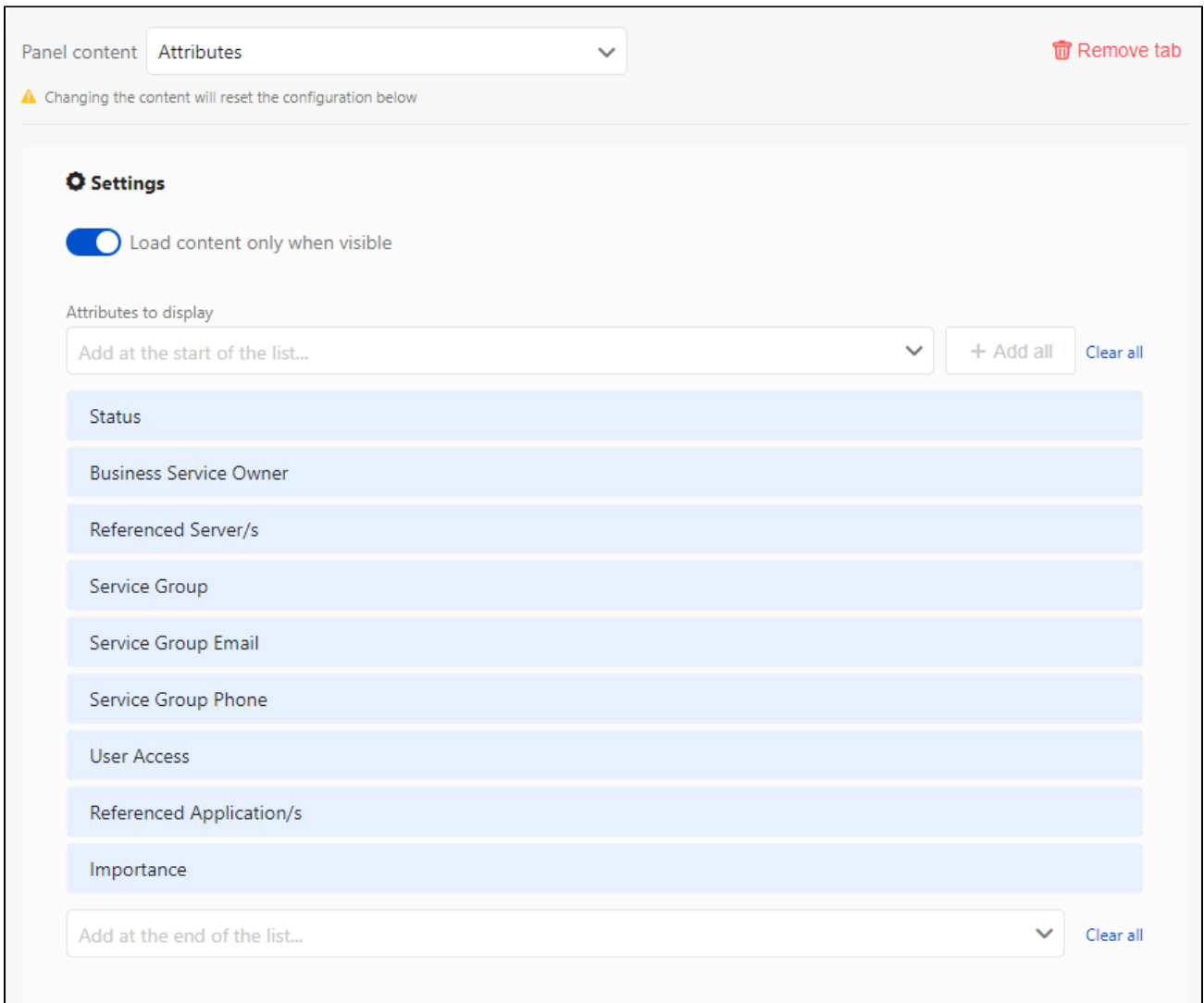
Adding widgets

Widgets are flexible components that you can insert in different places, such as object type layout page or dashboard. They can help you and your team quickly access information about your objects.

To add new widgets, or edit the settings for existing ones, go to your object schema, go to **Object type > Configure > Configure page layout**.

Attributes widget

Displays the selected attributes of your object in a customized order. It's useful to show specific information to your team without displaying all attributes at once, overwhelming them with data.



- **Add:** Select attributes from the drop-down list and add them to your widget.
- **Delete:** Hover over an attribute, and delete it.
- **Reorder:** Drag and drop attributes to change their order in the widget.

Settings

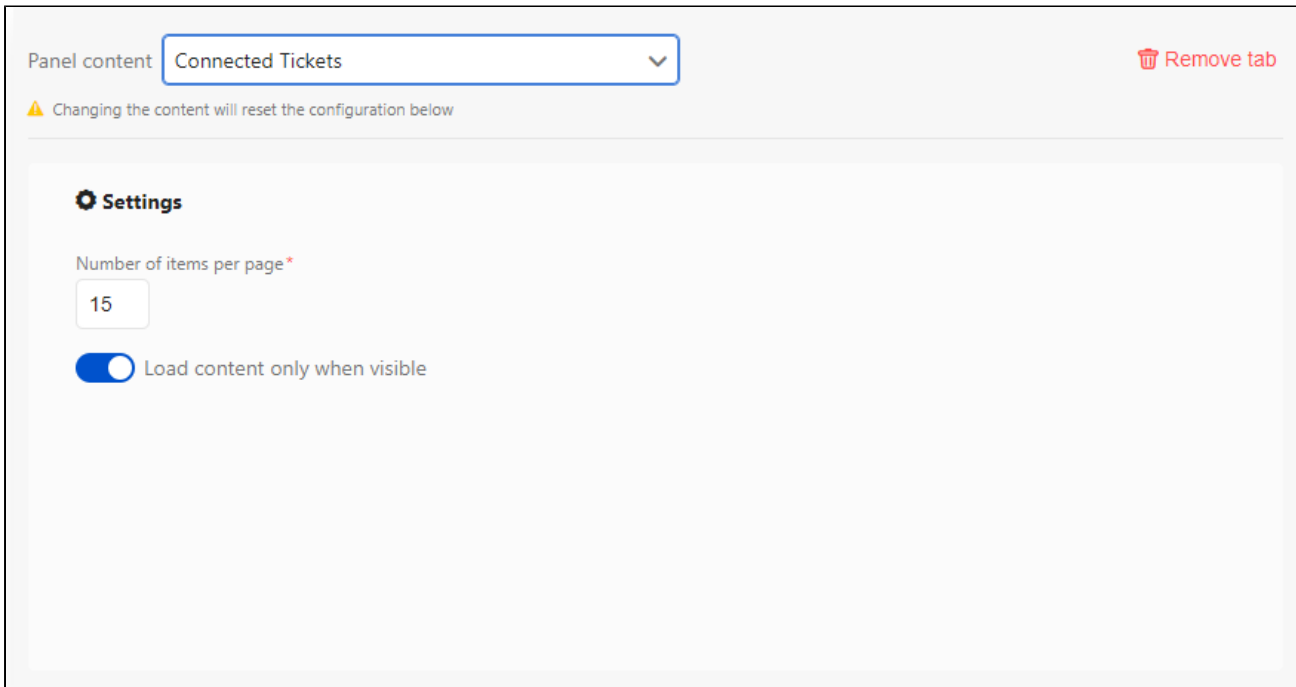
Settings for the Attribute widget:

Setting	Description
---------	-------------

<p>Display content in a collapsible panel</p>	<ul style="list-style-type: none"> • If ON, displays the attributes in a collapsible panel. • If OFF, displays the attributes inline with other page text. Automatically turns "load content only when visible" OFF, since the attributes always displayed. <p><i>This setting available in the side panel of the object page layout screen ONLY.</i></p>
<p>Load content only when visible (Lazy loading)</p>	<ul style="list-style-type: none"> • If ON, loads the data in the widget only when the widget is displayed. Default setting. • If OFF, loads the data in the widget with the page.

Connected tickets widget

Displays all Jira issues that are connected to an object.



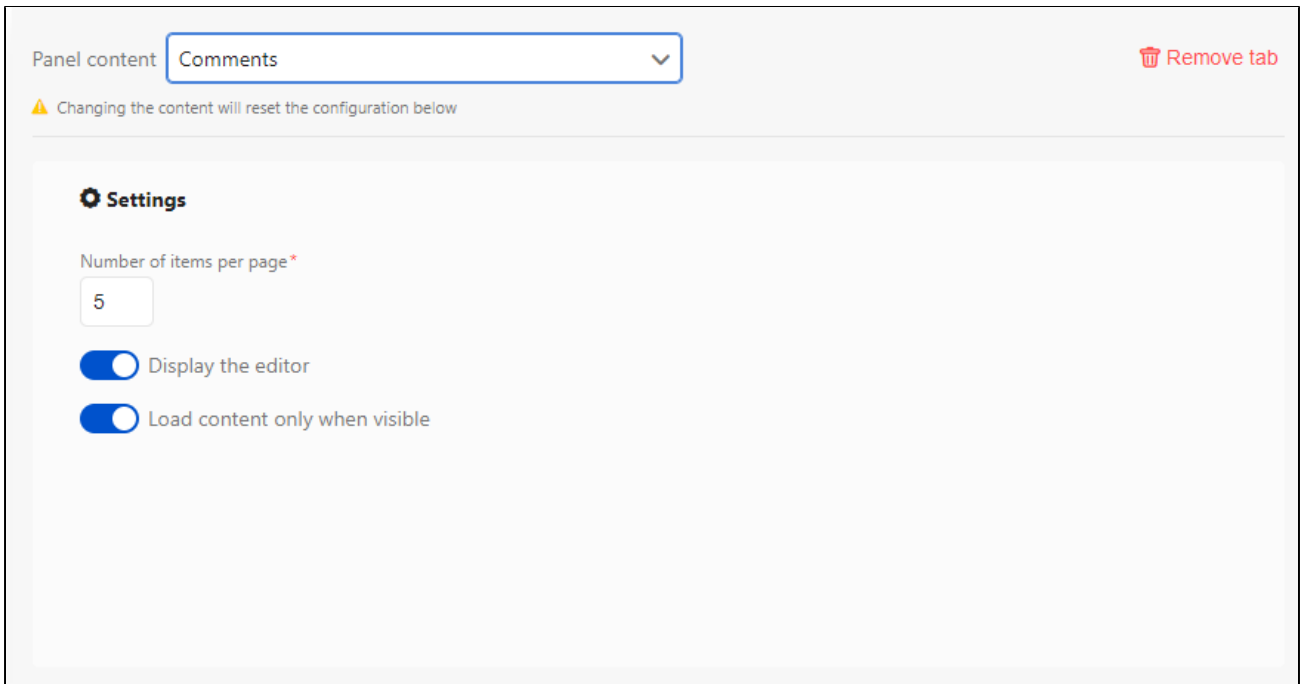
Settings

Settings for the Connected tickets widget:

Setting	Description
Number of items per page	Set the number of connected tickets displayed on each page.
Load content only when visible (Lazy loading)	<ul style="list-style-type: none"> • If ON, loads the data in the widget only when the widget is displayed. Default setting. • If OFF, loads the data in the widget with the page.

Comments widget

You can use this widget to view, edit, and add comments to an object.



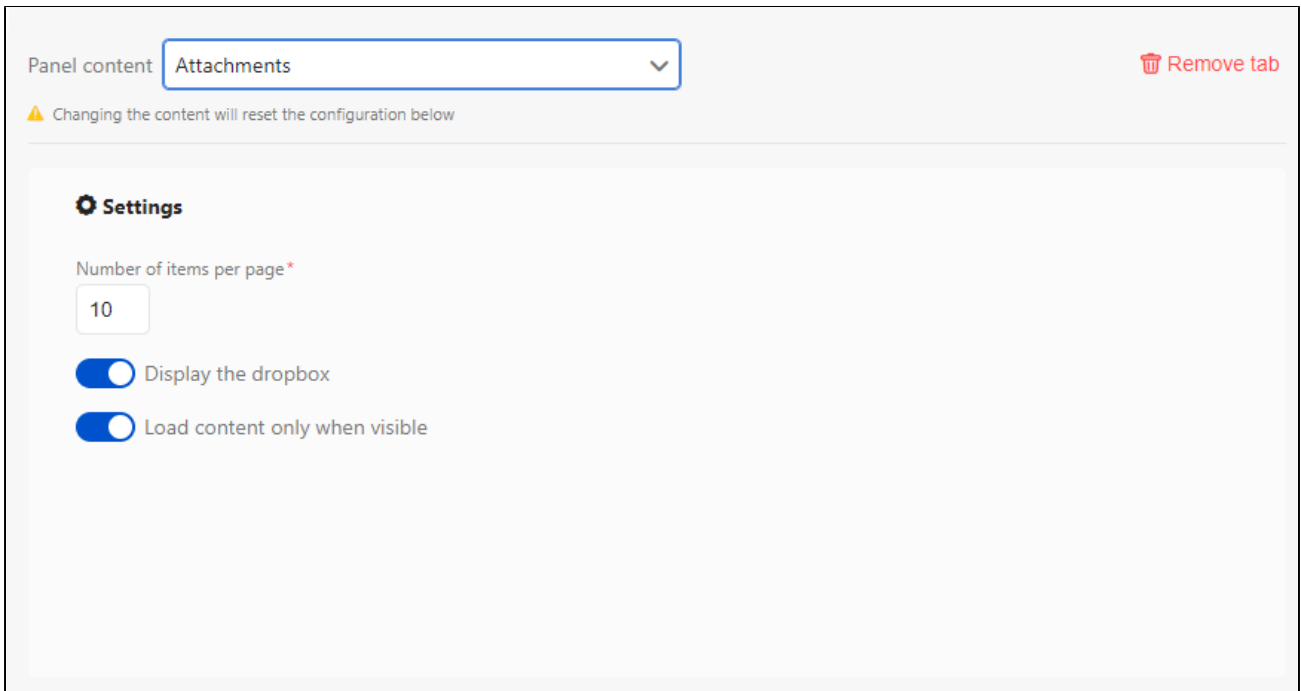
Settings

Settings for the Comments widget:

Setting	Description
Number of items per page	Set the number of comments displayed on each page.
Display the editor	<ul style="list-style-type: none"> • If ON, display the editor for comments at the top of the widget. • If OFF, does not display the widget editor.
Load content only when visible (Lazy loading)	<ul style="list-style-type: none"> • If ON, loads the data in the widget only when the widget is displayed. Default setting. • If OFF, loads the data in the widget with the page.

Attachments widget

You can use this widget to view, add, and delete attachments from an object.



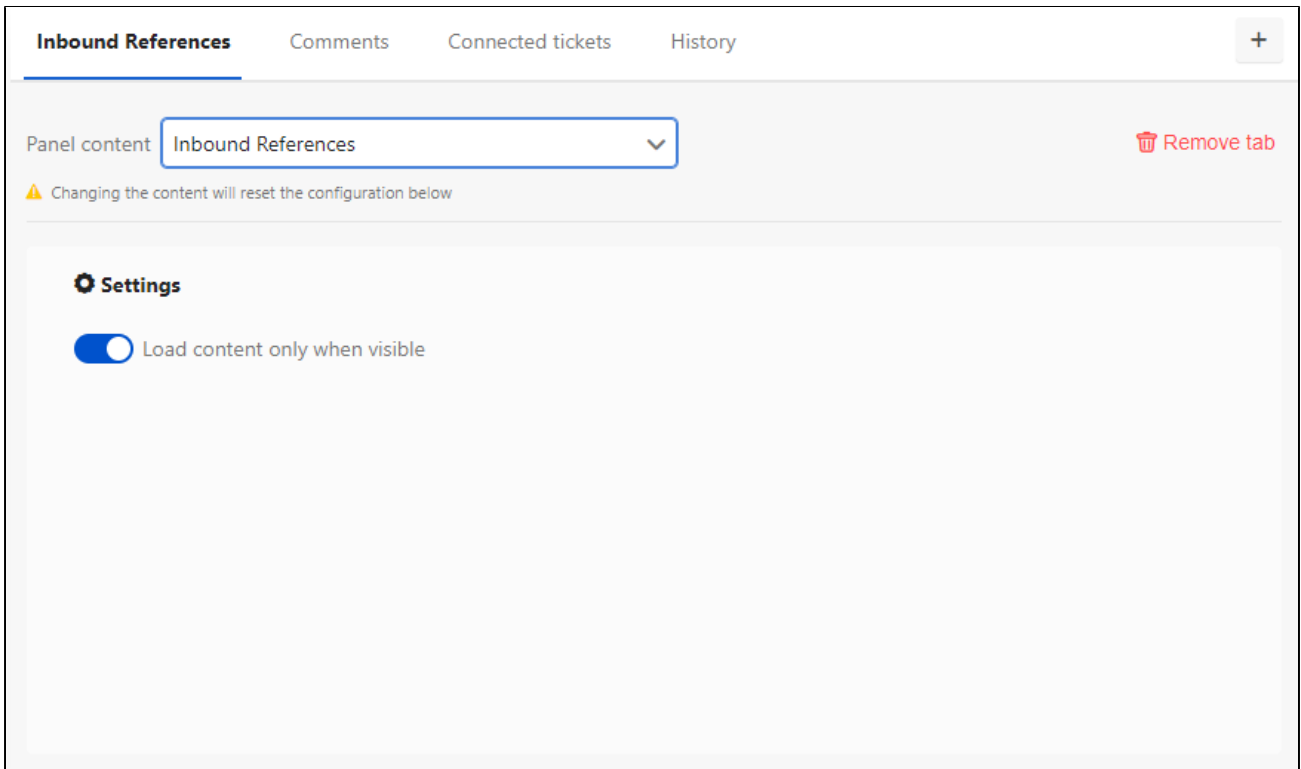
Settings

Settings for the Attachments widget:

Setting	Description
Number of items per page	Set the number of attachments displayed on each page.
Display the dropbox	<ul style="list-style-type: none"> • If ON, display the dropbox for adding attachments at the top of the widget. • If OFF, does not display the dropbox.
Load content only when visible (Lazy loading)	<ul style="list-style-type: none"> • If ON, loads the data in the widget only when the widget is displayed. Default setting. • If OFF, loads the data in the widget with the page.

Inbound references widget

You can use this widget to view inbound references to an object.



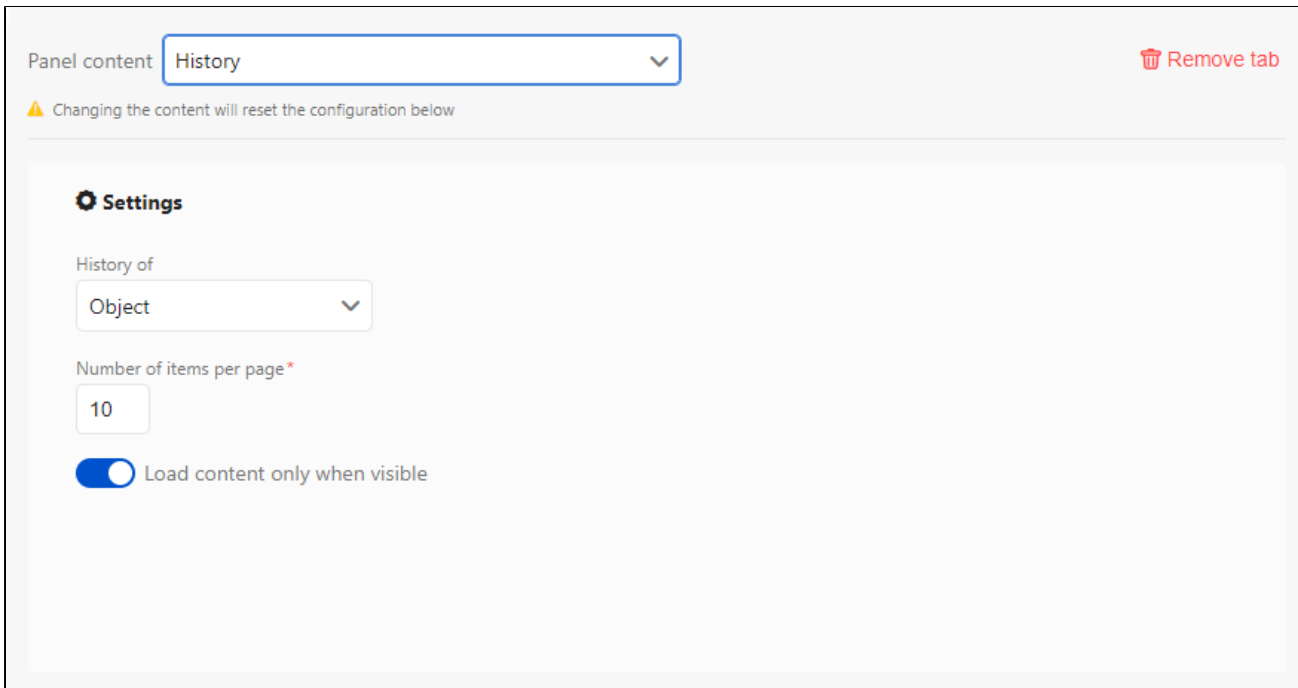
Settings

Settings for the Inbound references widget:

Setting	Description
Display content in a collapsible panel	<ul style="list-style-type: none"> • If ON, displays the inbound references in a collapsible panel. • If OFF, displays the inbound references inline with other page text. Automatically turns "load content only when visible" OFF, since the inbound references are always displayed. <p><i>This setting is available in the side panel of the object page layout screen ONLY.</i></p>
Load content only when visible(Lazy loading)	<ul style="list-style-type: none"> • If ON, loads the data in the widget only when the widget is displayed.Default setting. • If OFF, loads the data in the widget with the page.

History widget

You can use this widget to view the history of changes to an object, or the history of changes to all connected Jira issues.



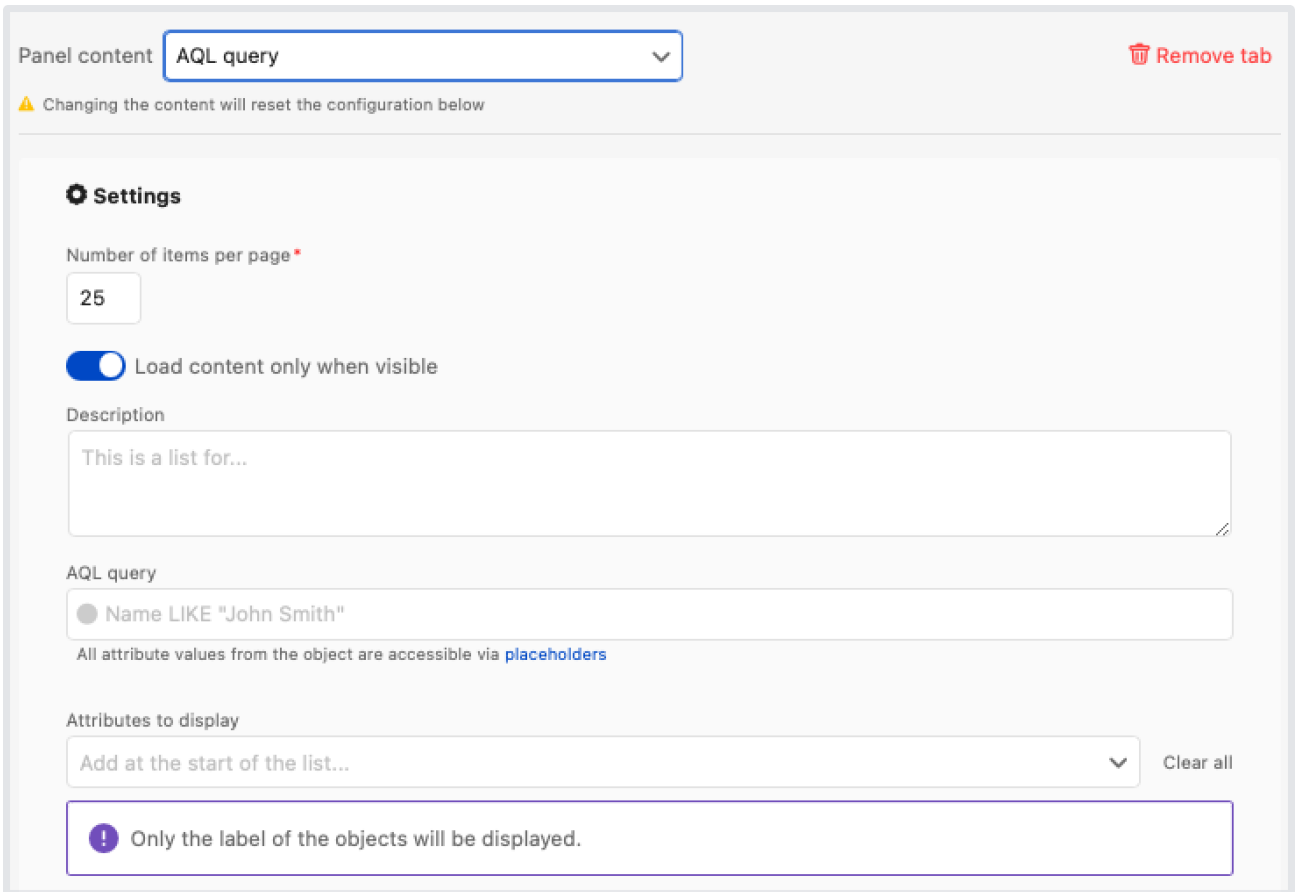
Settings

Settings for the History widget:

Setting	Description
History of	<ul style="list-style-type: none"> • If OBJECT is selected, displays the entire history of the object. • If CONNECTED TICKETS is selected, displays the history and changes to Jira tickets connected to the object.
Number of items per page	Set the number of history items displayed on each page.
Load content only when visible(Lazy loading)	<ul style="list-style-type: none"> • If ON, loads the data in the widget only when the widget is displayed. Default setting. • If OFF, loads the data in the widget with the page.

AQL query widget

You can use this widget to display objects, and their attributes, returned by an AQL query.



Settings

Settings for the AQL query widget:

Setting	Description
Number of items per page	Set the number of objects returned by the AQL query.
Load content only when visible(Lazy loading)	<ul style="list-style-type: none"> If ON, loads the data in the widget only when the widget is displayed. Default setting. If OFF, loads the data in the widget with the page.
Description	Enter a description of this query and the information returned.
AQL query	Enter an AQL query to run in the widget. For more information about AQL, see AQL - Assets Query Language .

Placeholders

Placeholders are very handy when used in the AQL query widget, because they can return information that is filtered by the current object. For example, the AQL query on the screenshot below returns all objects that meet the following criteria:

- Have outbound references to the current object
- Are **Network interfaces**

This is a great way of finding more detailed information about objects within an object type.

Panel content Remove tab

⚠ Changing the content will reset the configuration below

Settings

Number of items per page *

Load content only when visible

Description

AQL query

 `object HAVING outboundReferences(Key = ${Key}) and objectType = "Network Interface"`

All attribute values from the object are accessible via [placeholders](#)

Attributes to display

 Clear all

! Only the label of the objects will be displayed.

- For more info about Assets Query Language (AQL), see [Advanced search: AQL](#).
- For more info about placeholders, see [Advanced: Placeholders](#).

Working with objects

Objects are your actual assets. Every object is a digital representation of anything that you're mapping in Assets, be it a specific computer, employee, office they work at, or even a license for your software. You can create as many objects as you like, and group them within object types that represent their characteristics (an employee wouldn't be the best fit for the hardware object type).

Here's an example of an employee:

The screenshot shows a user interface for an employee object. At the top left is a profile picture and the text 'TUTISM-595 Anton Dry'. To the right are buttons for 'Graph', a print icon, 'Edit', and a menu icon. Below this are buttons for 'Attachment' and 'Comment'. A navigation bar contains 'Attributes', 'Connected tickets', 'Comments', and 'History'. The main content area lists attributes with their values:

Role	Developer
Team	Developers
Manager	Fred Astair
T-shirt size	Men Large
JIRA User	Anton Dry
Computer	Latitude 7480 D1
Mobile Phone	iPhone SN: 12345678
Active	✓
Office	London

References between objects

Information about an employee can be useful, but viewing their dependencies is even better. Thanks to this, you can know right away the office they work at, hardware they're using, role they belong to, or mobile phone they've been assigned. You can view references using the graphical viewer that lets you see all of it at a glance.

Here are the references of our example employee:

>> Anton Dry

Key: [TUTITSM-595](#)

Name: [Anton Dry](#)

Created: 26/Jan/21 4:11 PM

Updated: 26/Jan/21 4:11 PM

Role: [Developer](#)

Team: [Developers](#)

Manager: [Fred Astair](#)

T-shirt size: [Men Large](#)

JIRA User: [Anton Dry](#)

Computer: [Latitude 7480 D1](#)

Mobile Phone: [iPhone SN: 12345678](#)

Active: [true](#)

Office: [London](#)

Attachments

There are no attachments

Connected Jira issues

No related issues found for specified filter!

Get started with objects

Here's a list of pages to get you started:

- [Creating objects](#)
- [Adding objects to Jira issues](#)
- [Viewing objects in the object view](#)
- [Viewing references in a graph](#)
- [Printing labels and QR codes](#)

Creating objects

Objects are your actual assets or configuration items. What an object looks like depends on the object type it belongs to. [Learn more about objects](#)

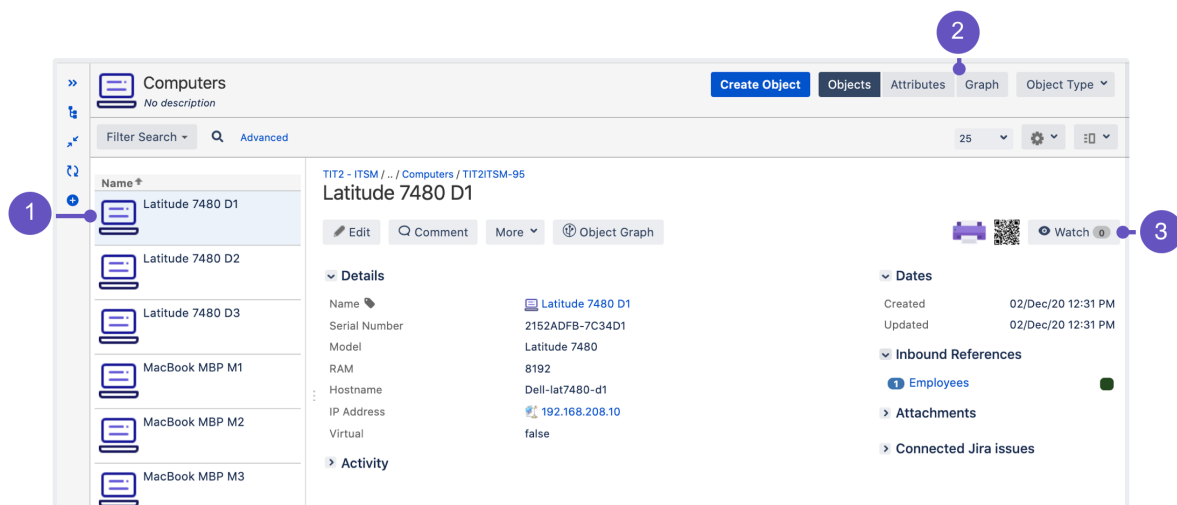
Create an object

To create an object:

1. Select the object type where you'd like to add your object. The object will use the object type's attributes, so that's quite important.
2. At the top-right, select **Create object**.
3. Enter the details and any required attributes.

Example object

Here's the view of an object, based on the sample ITSM schema:



1. **Object:**List of objects within an object type.
2. **Object type actions:**Actions for the parent object type.
3. **Object actions:**Some of the object actions: printing labels or QR codes, watching.

How object keys are generated

All objects have an object key. The key uses the pattern KEY-NUMBER where:

- KEY - object schema key
- NUMBER - sequence number from the Object table in the database

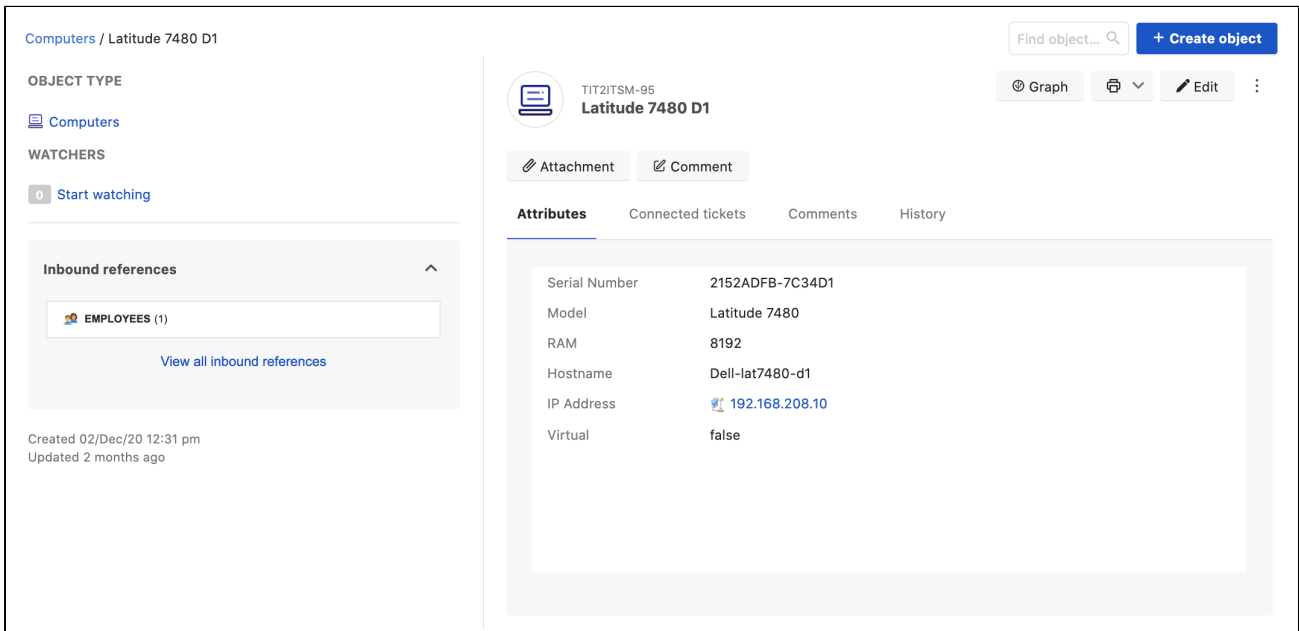
The KEY is used to categorize objects and easily identify the object schema they belong to. The NUMBER is the $\text{MAX}(\text{Object Id}) + 1$ from the Object table (AO_8542F1_IFJ_OBJ). A new object schema (e.g. CRM) could have its first object with a key of CRM-12345 if this is the number of the $\text{max}(\text{Object ID}) + 1$ received from the database.

In earlier versions, objects received keys based only on the object schema and the sequence in which they were created, for example KEY-1, KEY-2, KEY-3, and so on. This was changed for the following reasons:

- **Data Center support:** To ensure the keys are unique through the whole cluster, we switched to a database sequence.
- **Performance:** By using a database sequence, we can sync fewer steps while creating objects, which improves performance in a multi-threaded environment.

Using the object view

Here's the same object as seen in the object view:



The screenshot shows the Jira Service Management interface for an object. The breadcrumb is "Computers / Latitude 7480 D1". The object type is "Computers". The object ID is "TIT2ITSM-95" and the name is "Latitude 7480 D1". There are buttons for "Attachment" and "Comment". The "Attributes" tab is active, showing a table of properties:

Attribute	Value
Serial Number	2152ADFB-7C34D1
Model	Latitude 7480
RAM	8192
Hostname	Dell-lat7480-d1
IP Address	192.168.208.10
Virtual	false

Other visible elements include "Inbound references" showing "EMPLOYEES (1)", a "Start watching" button, and a search bar with a "+ Create object" button.

For more info on how you can customize the object view and what you can do in it, see [Viewing objects in the object view](#).

Next steps

Have a look at these pages:

- [Viewing objects in the object view](#)
- [Viewing references in a graph](#)
- [Adding objects to Jira issues](#)

Adding objects to Jira issues

When you add objects to your Jira issues, information about them will be displayed directly on the issues, giving everyone interested enough context. Depending on how your workflows are set up, the added objects can also be affected when an issue is moving through a workflow.

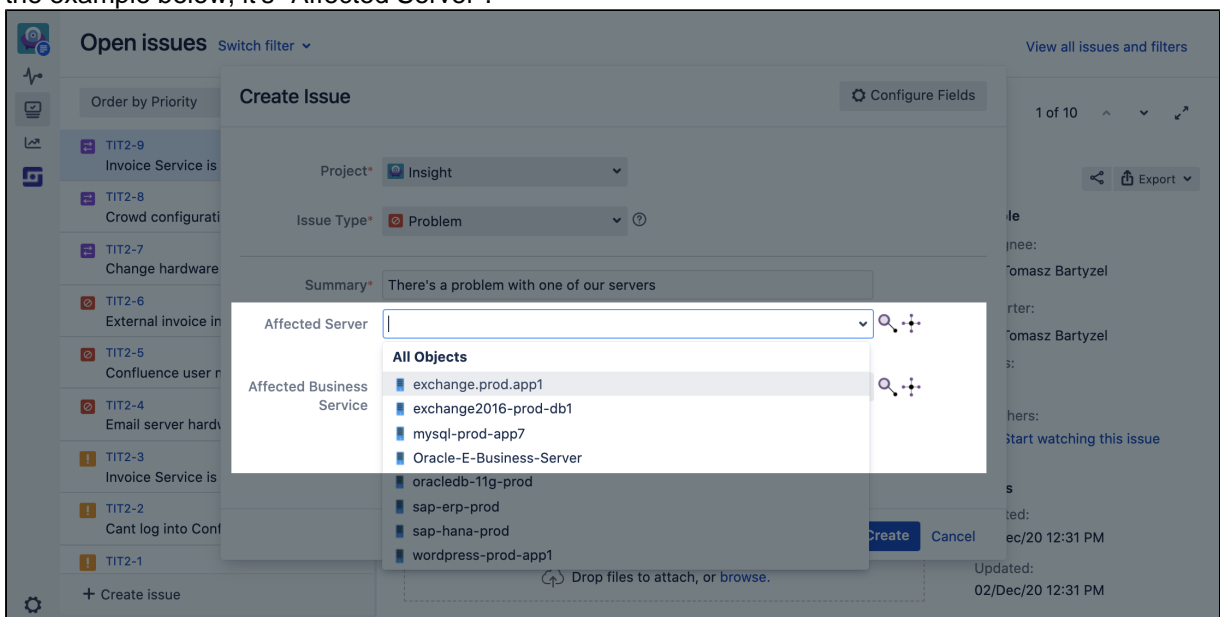
Before you begin

You'll add objects to your issues by choosing them as values of the Assets custom fields. These custom fields need to be first added to the right screens in Jira, which needs to be done by your Jira admin. For more info, see [Adding Assets custom fields to screens Jira](#).

Add an object to your Jira issue

To add an object to a Jira issue:

1. Create or edit a Jira issue, and locate the Assets custom field. It will look just like any other field. In the example below, it's "Affected Server".

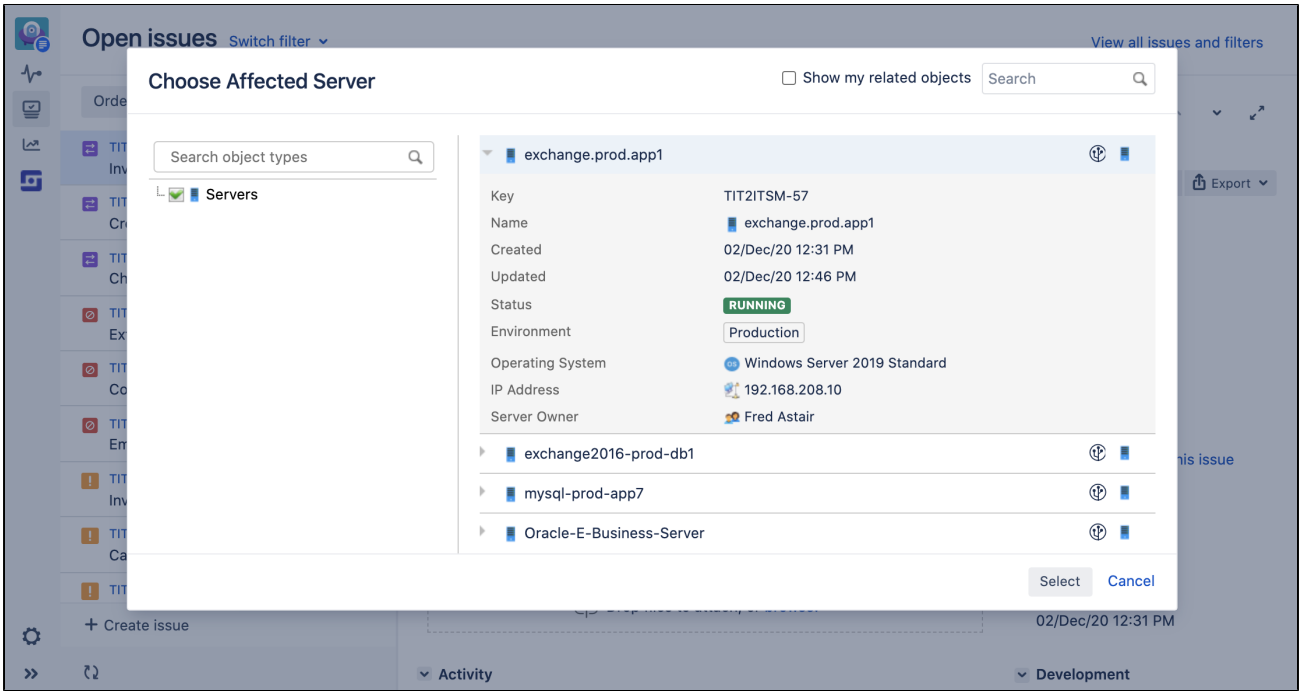


2. Select your object from the drop-down menu. If you can't find it, try using the object picker that will let you browse available objects.

Select objects with the object picker

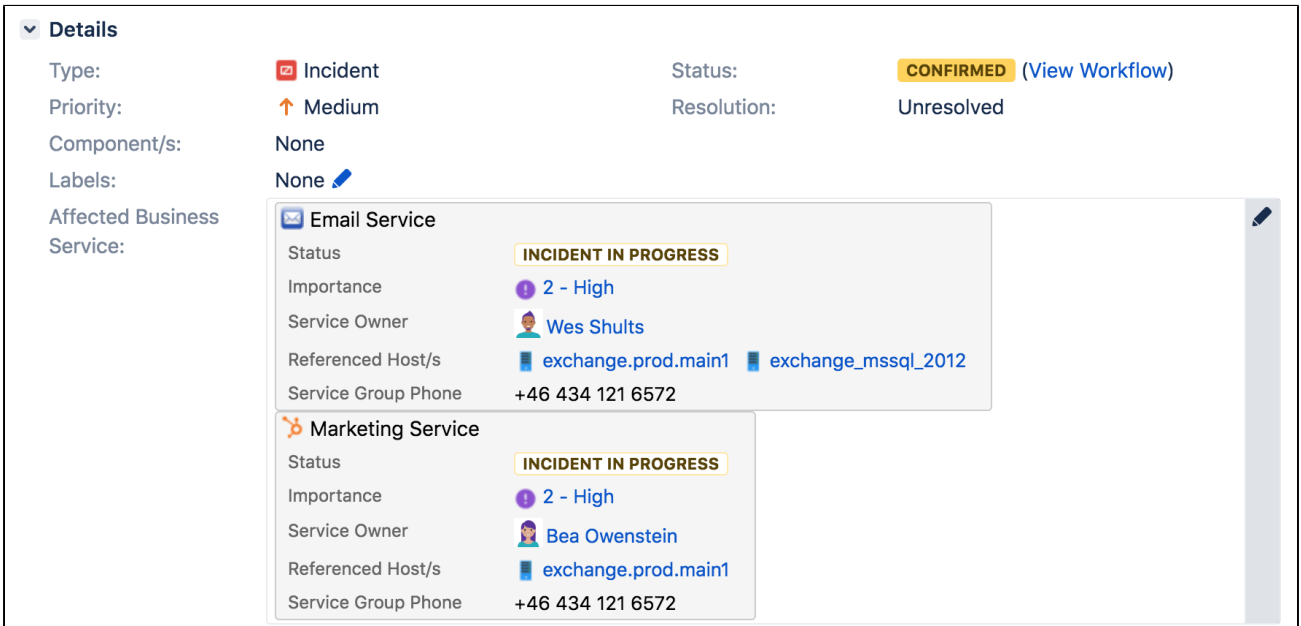
The object picker lets you search through **Object types** (left-side) and **Objects** (right-side).

- You can expand objects to view their attributes. This should help you make sure you're choosing the right one.
- You can select + Shift an object type to select all object types in its tree structure.



View object details on your Jira issues

All objects connected to an issue are shown when you view this issue. The amount of details shown depends on how the custom field (responsible for adding the object) is configured. Some custom fields might allow fewer attributes than others, and that's up to how your Jira admin configured them.



You can select an object to view all details about it. You can then use the icons to switch between overview, attributes, or graph.

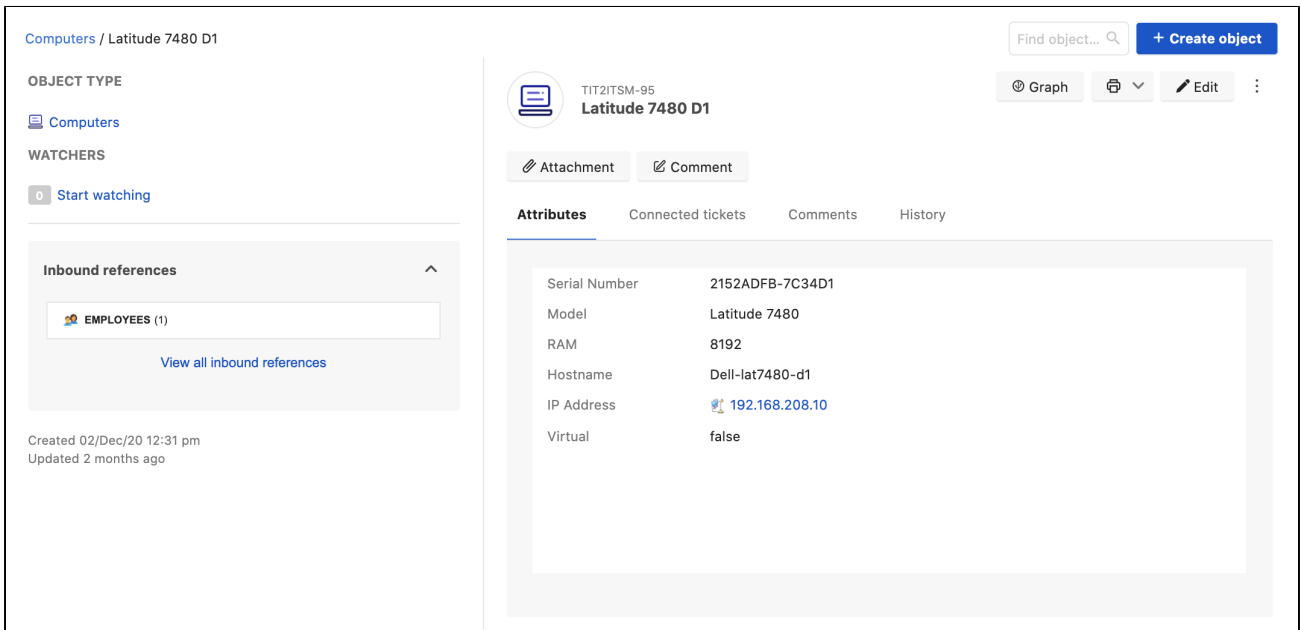
The screenshot shows the 'Object Overview' modal in Jira Service Management. The modal title is 'Object Overview' and it includes a search bar. The main content area displays details for the object 'exchange.prod.app1'. The details are as follows:

Property	Value
Key	TIT2ITSM-57
Name	exchange.prod.app1
Created	02/Dec/20 12:31 PM
Updated	02/Dec/20 12:46 PM
Status	RUNNING
Environment	Production
Operating System	Windows Server 2019 Standard
IP Address	192.168.208.10
Server Owner	Fred Astair

At the bottom right of the modal, there is a 'Close' button. The background shows a list of issues with columns for 'Description' and 'Development'.

Viewing objects in the object view

The object view lets you see all object details, including comments, attachments, and the history of changes. You can also view and modify object attributes, see all issues connected to an object, or view its references. All in a nice and friendly way.



Accessing the object view

You can access the object view in many ways, the easiest one being just selecting the object key (displayed as breadcrumbs when viewing the object):

- Select **Assets > Recent objects**, and select any object.
- Select any object in the schema and select its object key (displayed in breadcrumbs above the project name)
- [Search for objects](#) and select the object key.

Customize the object view

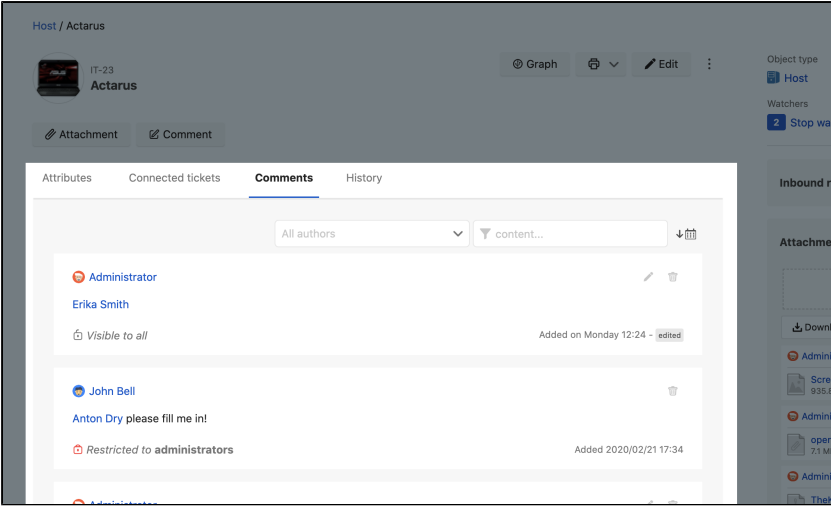
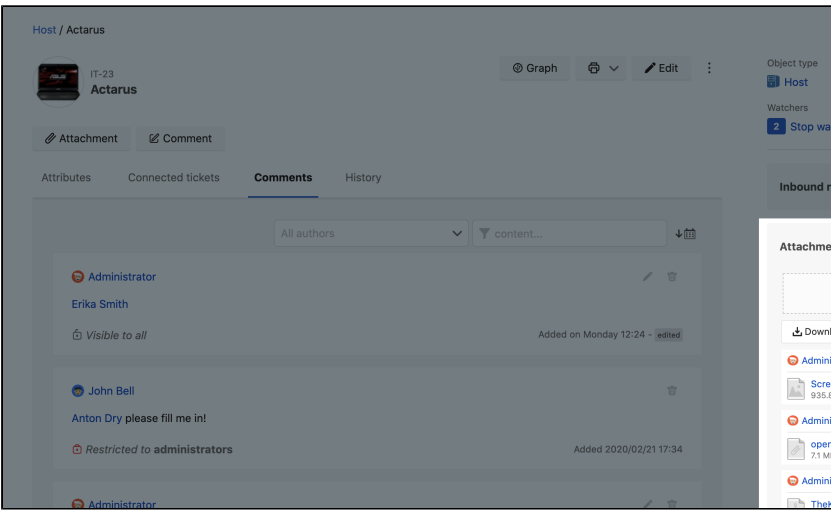
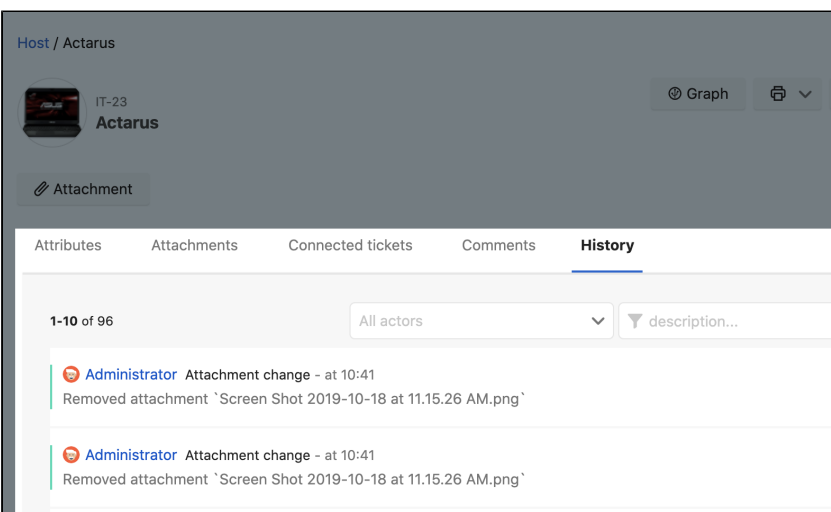
If you have the permission to edit the object schema, you can customize the object view to only display the information you need. For more info, see [Customizing the object view layout](#)

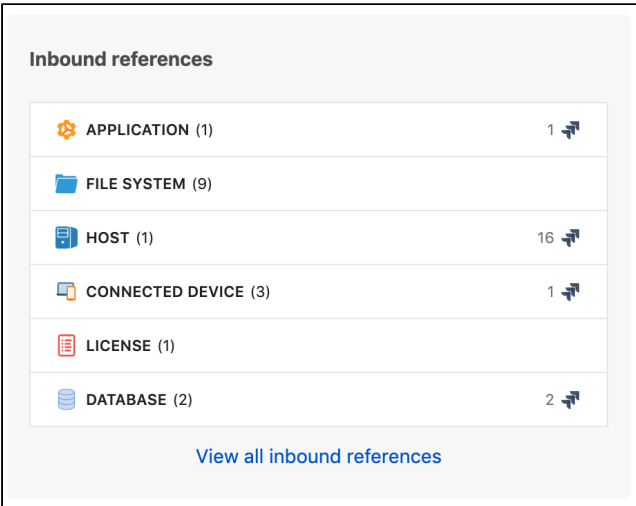
What you can do in the object view



Here are the actions that you can perform on objects in the default object view:

Action	Description	Image
--------	-------------	-------

<p>1</p>	<p>View object details</p>	<p>The top section of the object view tells you about the object and where it first in your object schema. Here you can also view the object type, watchers, and the created or updated dates.</p>	
<p>2</p>	<p>View attributes</p>	<p>The Attributes tab contains information about all of the object's attributes. Select on attributes that are references to other objects to view those objects in more detail.</p>	
<p>3</p>	<p>View connected issues</p>	<p>The Connected ticket tab shows issues connected to the object. You can view all connected tickets, use the Unresolved toggle to limit them to unresolved tickets, or filter the tickets using any of your existing Jira filters. You can also filter the tickets by summary or description.</p>	

<p>4</p>	<p>View comments and attachments</p>	<p>The Comments tab lets you view and add comments for the object. You can filter them by author, description, or summary. You can add comments by selecting Comment.</p>	
<p>5</p>	<p>View attachments</p>	<p>You can see attachments added to the object in the attachments widget on the right.</p>	
<p>6</p>	<p>View history of changes</p>	<p>The History tab shows a time-ordered list of changes made to this object.</p>	

<p>7</p>	<p>View references in a graph</p>	<p>After selecting Graph at the top-right, you can view references between this and other object within the object schema.</p>	
<p>8</p>	<p>View inbound references</p>	<p>In the Inbound references widget, you can see all objects that have some references to this object.</p>	
<p>9</p>	<p>Edit, move, or delete objects</p>	<p>If you are an Assets Schema administrator, you can edit an object using the Edit button. You can also move an object to a different object type, or delete it entirely.</p>	

<p>10</p>	<p>Print labels or QR codes</p>	<p>If you are an Assets Schema manager or Assets Schema administrator, you can also print a label or a QR code for this object.</p>	<div data-bbox="644 188 831 226"> <p>Host / Actarus</p> </div> <div data-bbox="644 271 772 405">  </div> <div data-bbox="788 302 916 374"> <p>IT-23 Actarus</p> </div> <div data-bbox="644 450 895 510"> <p> Attachment</p> </div> <div data-bbox="667 562 1382 595"> <p>Attributes Attachments Connected tickets</p> </div>
------------------	--	---	---

Viewing references in a graph

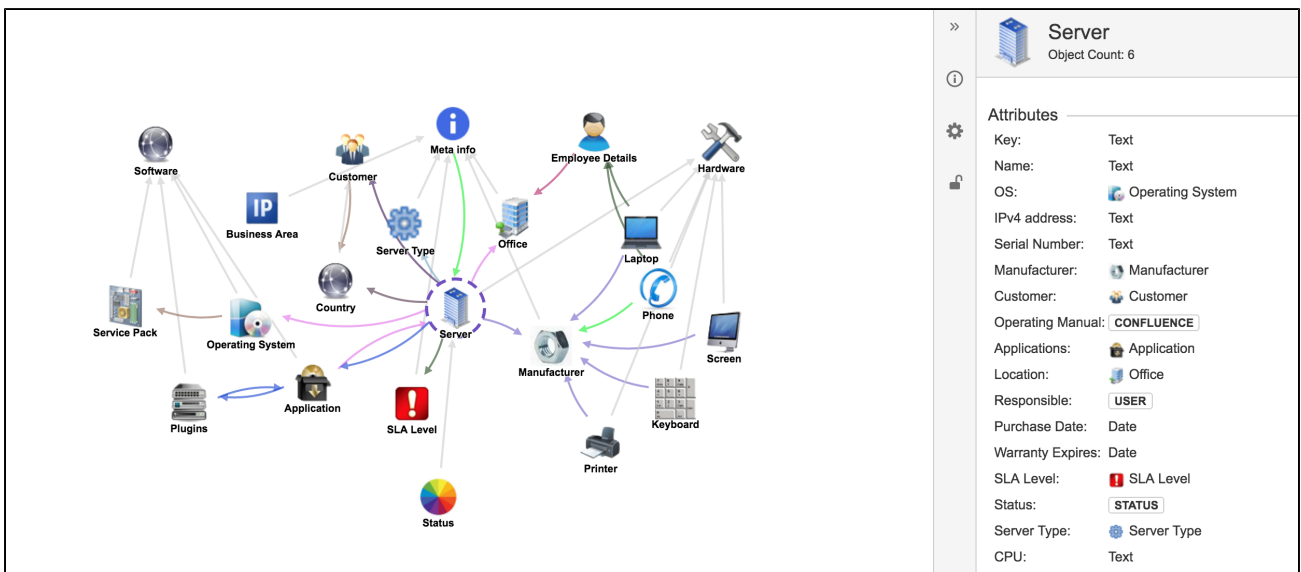
Assets contains four different graphs to help you get a better overview of your data:

- Object schema graph
- Object type graph
- Object graph (including force graph and tree graph)

Object schema graph

To view an object schema's graph:

1. Open your object schema.
2. At the top-right, select **Graph**.



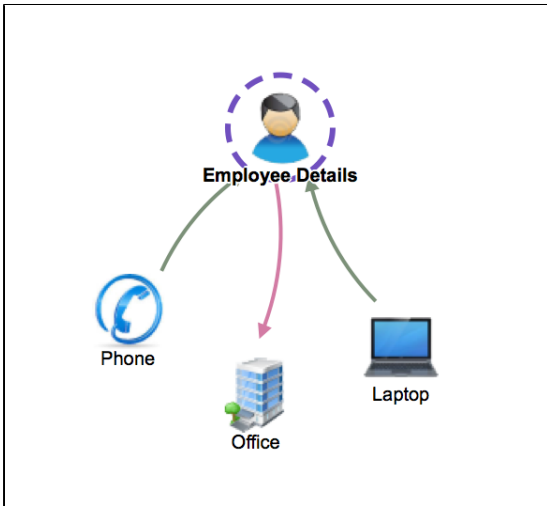
To get information about a certain object type, click on it.

- Gray lines point to the parent object type.
- Colored lines mark object attribute dependency.
- You can click the cog icon in the vertical menu to hide parent lines or visualize the object count.
- You can also create new references directly in the graph by pressing shift and dragging from one object type to another indicated by a purple line as seen in the image below.

Object type graph

To view an object type's graph:

1. Open your object type.
2. At the top-right, select **Graph**.



The object type graph is mostly the same as the object schema graph, but allows you to focus more on a specific part. By double-clicking on the icon for an object type, you can expand the graph and also show the types connected to this one.

Object graph

To view an object's graph.

1. Select your object.
2. Select **Object graph**.

Force graph view

The object graph is used to view specific objects and their relations. On the right, you can see all attributes, issues, and attachments for the selected object. By double-clicking an object, you can expand the graph to include all related objects.

View Object Graph

>> **Acme Integration. Server**

ⓘ

⚙️

🔒

Details

Key: ASSET-102

Name: Acme Integration. Server

OS: Ubuntu

IPv4 address: 223.321.223.221

Serial Number: 982343922

Manufacturer: HP

Customer: NASA

Operating Manual: [Unknown Confluence Page \(3506200\)](#)

Location: Stockholm

Responsible: Tommy Nordahl [Riada]
Luke Skywalker

Purchase Date: 10/Mar/15

Warranty Expires: 29/Apr/16

SLA Level: Level 3

Status: RUNNING

Close

As with the other graphs you can stop the icons moving around by clicking the lock icon. In the configuration, you can set the reference depth which tells how far from the initial object the graph should show, i.e. if you set it to 1 you will only see the selected object and related objects, set it to 2 and you will also see all objects related to the related objects and so on.

Tree graph view

This view let you watch your object references in a hierarchical tree structure. Referenced objects are grouped by Object Type to make it easier to distinguish them.

View Object Graph

Force Tree

Computer

ASUS Zenpad 10

MacBook Pro

Phone

iPhone 6

Office

Stockholm

Staff

Mathias

Referenced objects

Filter on an additional IQL...

- ASUS Zenpad 10
- MacBook Pro

Key: COM-17
Name: MacBook Pro
Owner: Fredrik
Created: 06/Nov/17 10:24 PM
Updated: 02/Feb/18 11:11 AM

Close

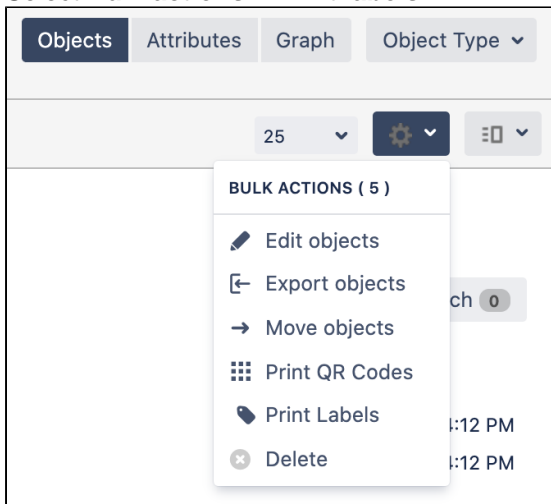
Printing labels and QR codes

Using labels and QR codes can really streamline the management of tangible assets. For this, Assets lets you print labels and QR Codes for any object. These documents show you how to do that.

Print labels

The easiest way to print labels is to search for multiple objects and then use bulk actions to print labels for all of them.

1. Search for your objects or select an object type (you can also print individual labels by selecting the printer icon on an object).
2. Select **Bulk actions > Print labels**.



3. If there's a template available, select it, and select Print. Otherwise, your admin will need to create a template.

Here's an example of a ready label:



Print QR codes

You can view an object's QR code after opening your object.



To print QR codes for multiple objects:


1. Search for your objects or select an object type (you can also print individual QR codes as shown on the image above)
2. Select **Bulk actions > Print QR codes**.

Importing your data

Entering everything manually could be a lifes work in a large organization. So thats why there are a few tools to help.

Import	Description
Asset Discovery	Asset Discovery is an agentless scanner that picks up network assets. You can choose which of assets, and which attributes, you pull into your Assets object schemas and you can create your own scanning patterns to find more specific details See Importing with Asset Discovery
Integrations	Integrations can be used to connect to other tools, such as cloud services, asset managers, and other CMDBs. These integrations can be downloaded for free from Atlassian Marketplace, and include AWS, Azure, Google Cloud, ServiceNow, and some more. See Importing with Integrations
Built-in importers	Import your data from CSV, database, Jira users, JSON, LDAP, or other object schema. Once you create your import configuration, you can keep it in sync so your assets stay up to date. See Importing with built-in importers

Importing with Asset Discovery

 Asset Discovery is available for free for Jira Service Management Data Center.

Asset Discovery is an agentless scanner (although there is an agent available to get more detailed information) that picks up network assets. You can select which of assets, and which attributes, you pull into your Insight object schemas and you can create your own scanning patterns to find more specific details. If you run it on a schedule it will pick up changes to keep data updated. With automation rules you can even trigger Jira issues, email notifications, and more based on detected changes.

- [Download Asset Discovery from Marketplace](#)
- [View Asset Discovery documentation](#)

Importing with Integrations

Asset Integrations can be used to connect to other tools, such as cloud services, asset managers, and other CMDBs.

While we have all these tools, we don't recommend you bring in every bit of data you have into Assets unless you plan to depreciate the tool. Bring in what you need to use in Jira Service Management, you can always bring in more later.

 All Asset Integrations can be installed through the Marketplace for free.

Here is the full list of Asset Integrations:

- [Asset Integration - AWS](#)
- [Asset Integration - Azure](#)
- [Asset Integration - Google Cloud](#)
- [Asset Integration - SCCM](#)
- [Asset Integration - Snow](#)
- [Asset Integration - ServiceNow](#)
- [Asset Integration - Jamf](#)
- [Asset Integration - Device42](#)
- [Asset Integration - NVD](#)
- [Asset Integration - Jira & Bitbucket](#)
- [Asset Integration - Tempo](#)

Importing with built-in importers

Assets has several built-in importers that let you import your data from CSV, database, JSON, LDAP, and so on. You'll use these importers by creating an import configuration, specifying its details, and then mapping the data you're importing to object types and attributes in Assets. Such an import configuration can be then synced on a regular basis, so your assets stay up to date.

 You need to be an Assets Manager for an object schema to create, configure, and enable imports.

Get started with importers

- [Understand importing concepts](#)
- [1. Create your import configuration](#)
 - [CSV import](#)
 - [Database import](#)
 - [Jira users import](#)
 - [JSON import](#)
 - [LDAP import](#)
 - [Object schema import](#)
- [2. Create object type and attribute mapping](#)
- [3. Inspect your import configuration](#)
- [4. Enable and sync your import](#)

Understand importing concepts

In Assets, you'll import your data by creating an import configuration, which consists of three elements:

- Base import configuration
- Object type mapping
- Attribute mapping (goes in pair with object type mapping)

Once created, such a configuration keeps importing your data (if synced) from updated data sources, and you can always make edits to it to make it even better. This page will help you understand the parts of your import configuration and the mappings you'll need to create. It's a good start before jumping right into importing your data.

i You need to be an Assets Manager for an object schema to create, configure, and enable imports.

Complete import configuration

Here's what a sample import configuration looks like after it's created in Assets. This should help you understand the concepts described on this page:

The screenshot shows the 'Employees' import configuration in Assets. It is divided into four numbered sections:

- Base import configuration:** This section is titled 'Employees' and 'CSV IMPORT'. It includes settings for 'Import from' (File), 'File' (computers.csv), 'Username' (No Value), 'Password' (No Value), 'Timeout' (No Value), 'Delimiter' (|), 'Encoding' (UTF-8), 'Automatically synchronize' (False), 'Synchronizing Account' (Tomasz Bartyzel), 'Empty Values' (Remove), 'Unknown Values' (Add), 'Concatenator' (-), 'Date Format' (No Value), and 'Date/Time Format' (No Value). A 'Create Object type mapping' button is visible.
- Object type mapping:** This section is also titled 'Employees' and 'ENABLED'. It includes settings for 'Selector' (No Value), 'Filter data source by IQL' (No Value), 'Missing objects' (Remove), 'Threshold Type' (Days), 'Threshold Number' (1), 'Empty Values' (Use default), 'Unknown Values' (Use default), and 'Match Identifier' (Case sensitive).
- Attribute mapping:** This section is a table with columns: Identifier, Data Locator, Insight Attribute, Object mapping (IQL), and Description. It shows two mappings: one for 'Name' (checked) and one for 'Computer' (unchecked).
- Another object type mapping (collapsed):** This section is titled 'Office' and 'ENABLED'. It is currently collapsed.

1. **Base import configuration:** That is from where, and when, will your data be collected. Your data source.
2. **Object type mapping:** Target object type for your data. You need to create a separate mapping for every object type that your data concerns.
3. **Attribute mapping:** Mapping between the data locators in your data source and Assets attributes. This mapping always goes together with the object type mapping, because these attributes are for your specific, target object type.
4. **Another object type mapping (collapsed):** Another object type mapping. It also has attribute mapping, when expanded. You can have as many object type mappings (and related attribute mappings) in one import configuration as you need.

Import configuration

Creating the base import configuration is the first step you do when you access **Import** in your object schema. When following the steps, you'll configure the following details:

- ✔ If you'd like to see the import types, available fields for each type, and the predefined structure and configuration, head to [1. Create your import configuration](#), and choose the import type.

Import types

Import type is the type of your data source. It can be CSV, database, Jira users, JSON, and so on. Depending on which import type you choose, the details you have to fill in will differ.

Import fields

Import fields are fields that define your import type. From simple name and description to concatenators and date formats, these fields will instruct Assets how to treat your data source. Import fields are specific to import types.

Scheduling

Details on whether your data source should be synced on a regular basis to keep the objects up to date. You can always enable it on the ready import configuration.

Predefined structure and configuration

The last step before creating your import configuration will be choosing whether you'd like Assets to create a predefined structure and configuration. These are optional and even if created can be modified after importing your data.

- **Predefined structure:** The import module will create an Assets object type structure, together with attributes, based on your data. This will only be available if you haven't created any object type mapping.
- **Predefined configuration:** The import module will create mappings between attributes and data locators, AQL references, and any other configuration items. Based on the name of the object type as well attribute names supplied from the import module, Assets will determine which object types(s) that should be added to the configuration. You can use the predefined configuration without using the structure.

Object type and attribute mapping

Assets needs to know where your data is going. Unless you created the predefined structure and configuration and are happy with it, you'll need to create object type and their related attribute mappings. Even with the predefined structure, you'll most likely need to make some edits and reviews.

- ✔ If you'd like to have a closer look at what fields are required when configuring the mappings and what they mean, head to [2. Create object type and attribute mapping](#).

A pair of object type and attribute mapping

Here's an example of an object type mapping and its attribute mapping. These will always go in pairs.

Employees
ENABLED
⚙️

Selector	No Value
Filter data source by IQL	No Value
Missing objects	Remove
Threshold Type	Days
Threshold Number	1
Empty Values	Use default
Unknown Values	Use default
Match Identifier	Case sensitive

Identifier	Data Locator	Insight Attribute	Object mapping (IQL)	Description
<input checked="" type="checkbox"/>	Name	Name		⚙️
<input type="checkbox"/>	Computer	Computer	Label = {Computer}	⚙️

In the example above, the mapping is specified for the Employees object type. It means that the attribute mapping will import data from your data source to the mapped attributes only for this object type. If your data is for multiple object types, you'll need to create such mappings for every one separately (still within the same import configuration).

This example is based on a simple CSV file that looked like this:

Source CSV file

```
Name,Computer
Alex,PC
Sara,Mac
```

The CSV headers (Name, Computer) are mapped to related Assets attributes (Name, Computer). "Name" was a text field, so it doesn't need additional object mapping (AQL). As for "Computer", it's a reference to another object type, so we used an [AQL](#) with [placeholders](#):

```
<Referenced object type identifier, Label> = {Header from the CSV file where Assets should look for data, Computer}
```

Here's a video that shows how this example was created:

Importing objects with references

Although importing data for default attributes (text, dates, etc) shouldn't be an issue, it might get tricky with data for objects that have references to other objects. It helps if you have created a structure of object types in Assets, because you know them, and can map them one by one. If you're having problems, see [Object type reference](#) for some examples.

Next steps

When you're ready, go to [1. Create your import configuration](#).

1. Create your import configuration

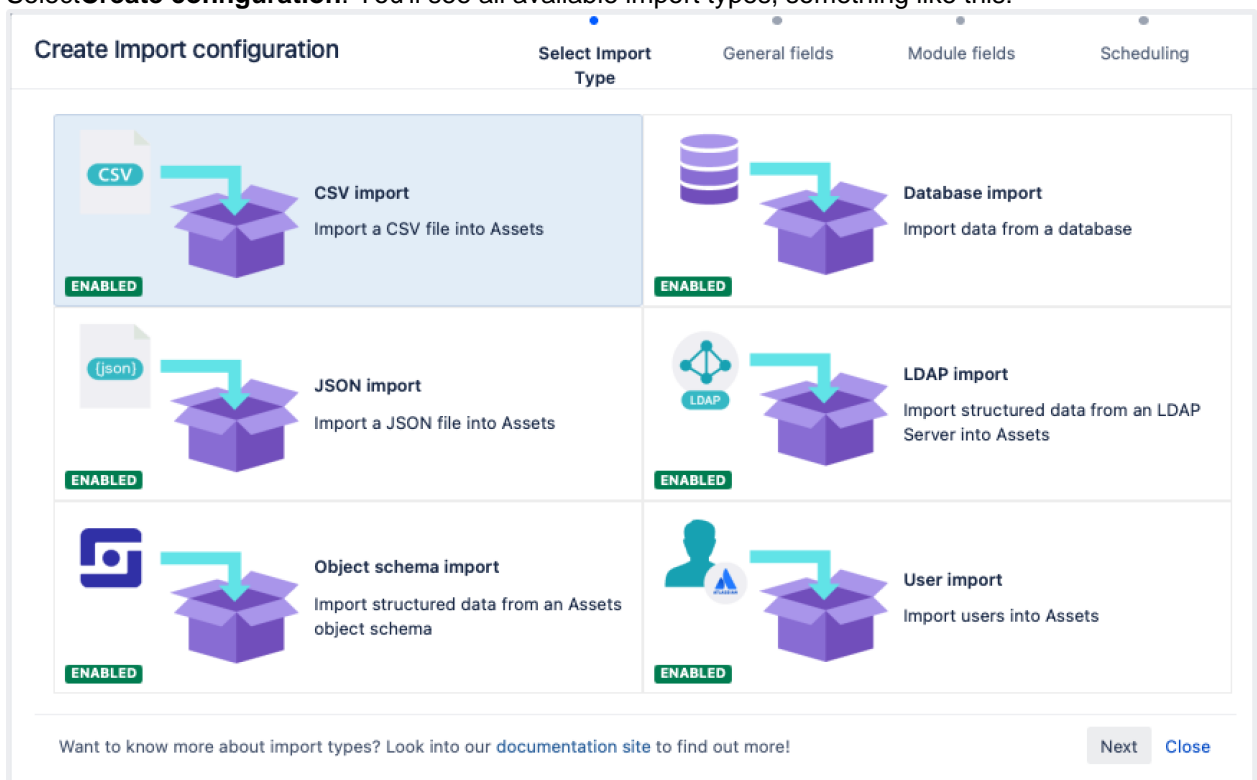
The first step to importing your data is creating an import configuration. The configuration depends on the import type (the source from which you're importing) and some of its configuration fields and options will differ based on what you've chosen. Assets provides some default import types available for configuration.

 You need to be an Assets Manager for an object schema to create, configure, and enable imports.

Create your import configuration

Your data will be imported into object schemas. To create a new configuration:

1. Open your object schema.
2. Select **Object schema > Configure**.
3. Switch to the **Import** tab.
4. Select **Create configuration**. You'll see all available import types, something like this:



Choose your import type

Once you've decided on your import type, select it and go to one of these pages:

- [CSV import](#)
- [Database import](#)
- [Jira users import](#)
- [JSON import](#)
- [LDAP import](#)
- [Object schema import](#)

More import types

When you install any of [Assets integrations](#), they will appear as a new import type. You can also [create your own import type](#), but this will require some developer knowledge.

CSV import

Your data will be imported from a CSV file. This is useful if you're still using spreadsheets (not judging) or have CSV files for all of your assets. [Learn more about importing](#)

i You need to be an Assets Manager for an object schema to create, configure, and enable imports.

Before you begin

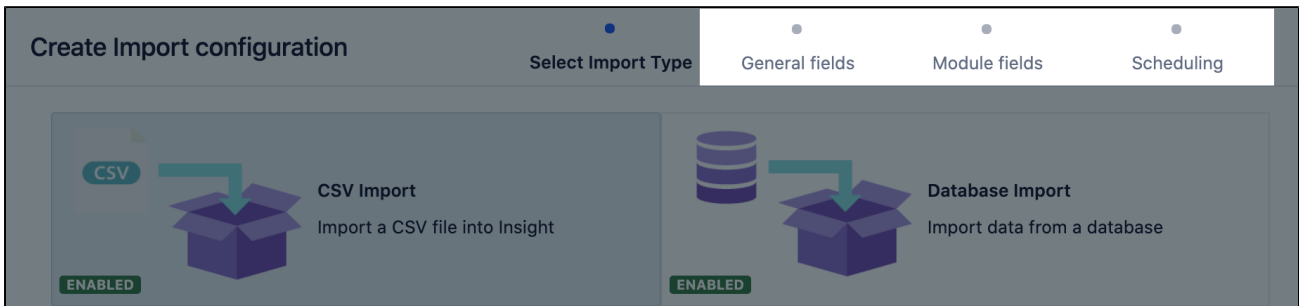
Here are some requirements for your CSV file:

The CSV file needs to have headers, as they are used to create the data locators for the imports. An empty header value is not allowed.

To divide multiple values (for the Assets Attributes with multiple cardinality), use || (2 pipe characters).

Import fields

Once you've chosen your import type, you'll need to enter details about it. Here's the description of fields you should see in Assets.



Here are general fields, common for every import type:

Name	Description
Name	The name of the import.
Description	The description for your convenience.
Concatenator	You can specify a default concatenator. When joining multiple data locators into one Insight attribute, this will be the default concatenator. One example could be to join two columns like "First name" and "Last name" into one attribute. So "Mathias" (first name) and "Edblom" (last name) will be concatenated as "Mathias Edblom" if using \s as concatenator. Enter \s for space-concatenated. To include a concatenate character, place the value between double quotes (i.e "\s").
Empty Values	Defines what should happen when a Data Locator is empty. Should the import remove the attribute value or just ignores it and leave the current value as is.
Unknown Values	Defines what should happen if a Data Locator is unknown to Insight. This could happen with attribute types like "Status" and "Select". The value can be added as an option or just ignore the value.
Date Format	Format for date fields in import source to convert dates into Insight. If left empty, Insight will automatically try to find correct format. The format should be specified according to the Java SimpleDateFormat guidelines .

Date/Time Format	Format for date/time fields in import source to convert dates into Insight. If left empty, Insight will automatically try to find correct format. The format should be specified according to the Java SimpleDateFormat guidelines .
-------------------------	--

These fields are specific to an import type (module).

Setting	Description
Import from	Choose a file or URL.
<ul style="list-style-type: none"> URL 	Protocol, Hostname, and Port of where to find the external CSV file. Example: <code>http(s)://example.com/file.csv</code> or <code>file:///opt/csv/example.csv</code>
<ul style="list-style-type: none"> File 	Choose a file to upload.
Username	Optional username for basic authentication.
Password	Optional password for basic authentication.
Delimiter	The delimiter used in the file. Enter <code>\t</code> for tab-delimited. To include a delimiter character, place the value between double quotes. To include a delimiter or a double quote character, place each value between double quotes.

Scheduling fields are responsible for keeping your data in sync:

Name	Description
Synchronizing Account	The Jira user to use when synchronize data into Insight
Cron Expression	The interval for the automatic synchronization.
Automatically Synchronize	If the import should be scheduled for automatic synchronization.

Pre-defined structure and configuration

In the next step, after you've filled in the required fields, Assets will ask you whether you want to create a predefined structure (object type mappings) and configuration (attribute mappings). Details of this will differ depending on the import type.

Here's some details for the CSV import:

- Each column from the CSV file will be created as an Assets attribute of type Default Text, and configuration will be mapped accordingly.
- The object type created will always be the same.
- Some object type mappings are disabled by default. Make sure to select the relevant ones.

If you want to have different attribute types created by the import of the data from the CSV file, create the predefined structure and then change the attribute types.

Import configuration created

You can now view your import configuration, but it's not ready yet. You still need to create or review the object type and attribute mapping, and make sure there are no problems with your import configuration.

When you're ready, go to [2. Create object type and attribute mapping](#).

Before you go


Object type mapping settings

In the next step, you'll create the object mapping settings. Here are some settings specific to the CSV import type.

Name	Description
Selector	The selector for the CSV import is not used. Use AQL to filter any data from the CSV file

Database import

You might want to import data from an internal or third-party system like BMC Remedy or Service Now. This article guides you through importing data from another database. [Learn more about importing](#)

 You need to be a Jira admin to create and configure database imports.

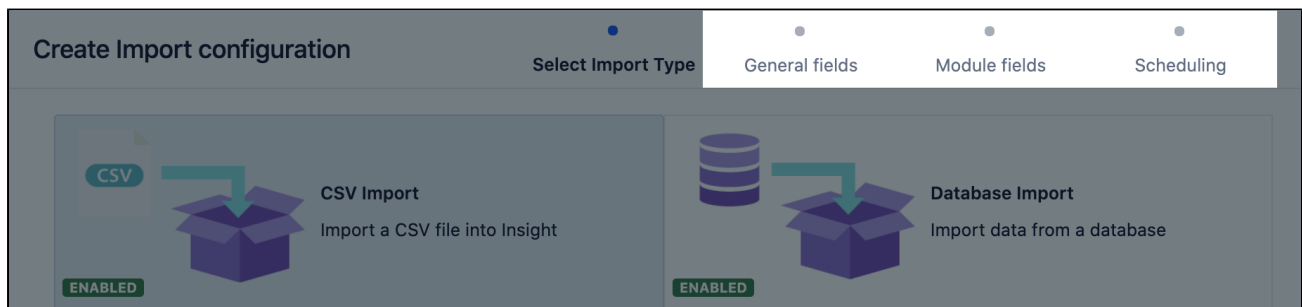
Supported databases

We're using a common framework (jdbc) to connect to the database, so we support many databases:

Database Type	Driver
Microsoft SQL Server	com.microsoft.sqlserver.jdbc.SQLServerDriver
Oracle	oracle.jdbc.OracleDriver
PostgreSQL	org.postgresql.Driver
MySQL	com.mysql.jdbc.Driver

Import fields

Once you've selected your import type, you'll need to enter details about it. Here's the description of fields you should see in Assets.



Here are general fields, common for every import type:

Name	Description
Name	The name of the import.
Description	The description for your convenience.
Concatenator	You can specify a default concatenator. When joining multiple data locators into one Insight attribute, this will be the default concatenator. One example could be to join two columns like "First name" and "Last name" into one attribute. So "Mathias" (first name) and "Edblom" (last name) will be concatenated as "Mathias Edblom" if using \s as concatenator. Enter \s for space-concatenated. To include a concatenate character, place the value between double quotes (i.e "\s").
Empty Values	Defines what should happen when a Data Locator is empty. Should the import remove the attribute value or just ignores it and leave the current value as is.
Unknown Values	Defines what should happen if a Data Locator is unknown to Insight. This could happen with attribute types like "Status" and "Select". The value can be added as an option or just ignore the value.

Date Format	Format for date fields in import source to convert dates into Insight. If left empty, Insight will automatically try to find correct format. The format should be specified according to the Java SimpleDateFormat guidelines .
Date/Time Format	Format for date/time fields in import source to convert dates into Insight. If left empty, Insight will automatically try to find correct format. The format should be specified according to the Java SimpleDateFormat guidelines .

These fields are specific to an import type (module).

Setting	Description
URL	The database connection URL. If you want to connect to the Jira database, you can look into the dbconfig.xml in Jira_HOME to find the correct configuration and URL.
Driver	The driver class to use to connect to the database. Place the JDBC driver (*.jar) to the <Jira_INSTALL>/lib folder Don't forget to restart Jira after adding the driver to lib folder.
Username	The username used to authenticate.
Password	The password used to authenticate. You can enter the password for your database.

If you want to query the Jira database, you can look into the dbconfig.xml found in Jira_HOME directory, to find out the correct configuration.

Scheduling fields are responsible for keeping your data in sync:

Name	Description
Synchronizing Account	The Jira user to use when synchronize data into Insight
Cron Expression	The interval for the automatic synchronization.
Automatically Synchronize	If the import should be scheduled for automatic synchronization.

Pre-defined structure and configuration

Predefined structure and configuration is not available for this import type.

Import configuration created

You can now view your import configuration, but it's not ready yet. You still need to create or review the object type and attribute mapping, and make sure there are no problems with your import configuration.

When you're ready, go to [2. Create object type and attribute mapping](#).

Before you go

In the next step, you'll create the object type and attribute mappings. Here are some settings specific to the database import type.

Object type mapping configuration

Name	Description
------	-------------

Selector The selector for the Database import is a SQL query. How to build SQL queries depends on the database type.

How to build up the selector (the SQL statement) depends on how you have configured the database connection. Below you will see some examples that will hopefully guide you in the right direction when setting this up.

Database Type	Example
Microsoft SQL Server	<p>select * from <database.name>.<schema-name>.Application</p> <p><schema-name> is "dbo" as default so an example could be:</p> <pre>select * from dbname.dbo.Application</pre>
Oracle	<p>select * from <schema-name>.Application</p> <p>If <schema-name> is "Jiraschema" the query will be:</p> <pre>select * from Jiraschema.Application</pre>
PostgreSQL	<pre>select * from "Application"</pre>
MySQL	<pre>select * from Application</pre>

If you want to query the Jira database, you can look into the `dbconfig.xml` found in `Jira_HOME` directory, to find out the correct configuration, such as `schema name` etc.

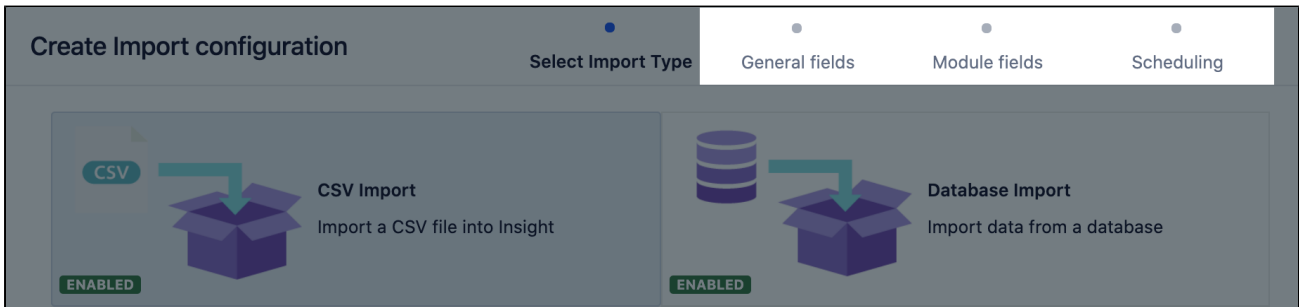
Jira users import

You might want to collect more information about your Jira users or relate Jira users to customers. For this, you need to import your Jira users or specific user groups in Assets. [Learn more about importing](#)

i You need to be an Assets Manager for an object schema to create, configure, and enable imports.

Import fields

Once you've chosen your import type, you'll need to enter details about it. Here's the description of fields you should see in Assets.



Here are general fields, common for every import type:

Name	Description
Name	The name of the import.
Description	The description for your convenience.
Concatenator	You can specify a default concatenator. When joining multiple data locators into one Insight attribute, this will be the default concatenator. One example could be to join two columns like "First name" and "Last name" into one attribute. So "Mathias" (first name) and "Edblom" (last name) will be concatenated as "Mathias Edblom" if using \s as concatenator. Enter \s for space-concatenated. To include a concatenate character, place the value between double quotes (i.e "\s").
Empty Values	Defines what should happen when a Data Locator is empty. Should the import remove the attribute value or just ignores it and leave the current value as is.
Unknown Values	Defines what should happen if a Data Locator is unknown to Insight. This could happen with attribute types like "Status" and "Select". The value can be added as an option or just ignore the value.
Date Format	Format for date fields in import source to convert dates into Insight. If left empty, Insight will automatically try to find correct format. The format should be specified according to the Java SimpleDateFormat guidelines .
Date/Time Format	Format for date/time fields in import source to convert dates into Insight. If left empty, Insight will automatically try to find correct format. The format should be specified according to the Java SimpleDateFormat guidelines .

These fields are specific to an import type (module).

Setting	Description
Restrict Groups	Only users in selected group(s) will be imported.

Include Inactive users	Specify if Assets should include inactive users from source.
-------------------------------	--

Scheduling fields are responsible for keeping your data in sync:

Name	Description
Synchronizing Account	The Jira user to use when synchronize data into Insight
Cron Expression	The interval for the automatic synchronization.
Automatically Synchronize	If the import should be scheduled for automatic synchronization.

Import configuration created

You can now view your import configuration, but it's not ready yet. You still need to create or review the object type and attribute mapping, and make sure there are no problems with your import configuration.

When you're ready, go to [2. Create object type and attribute mapping](#).

Before you go

In the next step, you'll create the object mapping settings. Here are some settings specific to the Jira users import type.

Object type mapping configuration

Name	Description
Selector	The selector for the Jira User import isn't used. User AQL to filter any data from the User data.

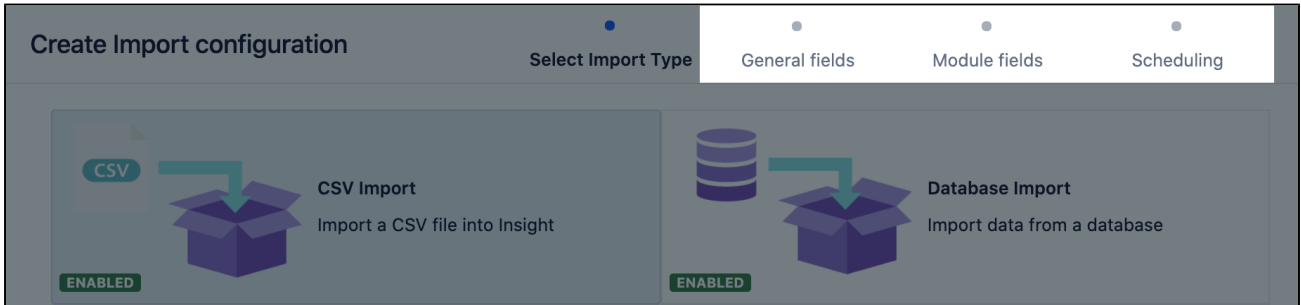
JSON import

This import type will import objects into Assets with a JSON file that holds the data to import. [Learn more about importing](#)

i You need to be an Assets Manager for an object schema to create, configure, and enable imports.

Import fields

Once you've chosen your import type, you'll need to enter details about it. Here's the description of fields you should see in Assets.



Here are general fields, common for every import type:

Name	Description
Name	The name of the import.
Description	The description for your convenience.
Concatenator	You can specify a default concatenator. When joining multiple data locators into one Insight attribute, this will be the default concatenator. One example could be to join two columns like "First name" and "Last name" into one attribute. So "Mathias" (first name) and "Edblom" (last name) will be concatenated as "Mathias Edblom" if using \s as concatenator. Enter \s for space-concatenated. To include a concatenate character, place the value between double quotes (i.e "\s").
Empty Values	Defines what should happen when a Data Locator is empty. Should the import remove the attribute value or just ignores it and leave the current value as is.
Unknown Values	Defines what should happen if a Data Locator is unknown to Insight. This could happen with attribute types like "Status" and "Select". The value can be added as an option or just ignore the value.
Date Format	Format for date fields in import source to convert dates into Insight. If left empty, Insight will automatically try to find correct format. The format should be specified according to the Java SimpleDateFormat guidelines .
Date/Time Format	Format for date/time fields in import source to convert dates into Insight. If left empty, Insight will automatically try to find correct format. The format should be specified according to the Java SimpleDateFormat guidelines .

These fields are specific to an import type (module).

Setting	Description
---------	-------------

Import from	Choose a file or URL.
• URL	Protocol, Hostname, and Port of where to find the external JSON file. Example: http(s)://example.com/rest/api/service or file:///opt/csv/example.json
• File	Choose a file to upload.
Username	Optional username for basic authentication.
Password	Optional password for basic authentication.
Encoding	The encoding used in the file.

Scheduling fields are responsible for keeping your data in sync:

Name	Description
Synchronizing Account	The Jira user to use when synchronize data into Insight
Cron Expression	The interval for the automatic synchronization.
Automatically Synchronize	If the import should be scheduled for automatic synchronization.

Pre-defined structure and configuration

In the next step, after you've filled in the required fields, Assets will ask you whether you want to create a predefined structure (object type mappings) and configuration (attribute mappings). Details of this will differ depending on the import type.

Here's some details for the Jira users import:

- The predefined structure and configuration will be created based on the JSON file/URL present in the configuration.
- Some object type mappings are disabled by default. Make sure to select the relevant ones.

Import configuration created

You can now view your import configuration, but it's not ready yet. You still need to create or review the object type and attribute mapping, and make sure there are no problems with your import configuration.

When you're ready, go to [2. Create object type and attribute mapping](#).

Before you go

In the next step, you'll create the object type and attribute mapping. Here are some settings specific to the JSON import type.

Object type mapping settings

Name	Description
------	-------------

Selector	<p>The selector for the JSON import.</p> <p>Use[Attribute.*[Attribute]]</p> <p>Example:</p> <ul style="list-style-type: none"> • "items" • "items.item" • "json.data.systems" <p>Only JSON arrays are valid as selector.</p>
-----------------	---


Attribute mapping settings

Data locator:

Name	Description
attribute1[*].attribite2 [*]..	The data locators provided are depending on the JSON and the selector configured.

LDAP import

Maybe you work with a corporate directory that contains your assets or employee-manager relationships used for approval processes. Such LDAP entries can be imported in Assets. To make things easy, Assets has modules that works with popular LDAP directories, which fetch the structure and the assets from your directory. This article shows you how to set this up.[Learn more about importing](#)

 You need to be a Jira admin to create and configure LDAP imports.

Overview

An LDAP directory is a collection of data about users and other assets. LDAP (Lightweight Directory Access Protocol) is an Internet protocol that web applications can use to look up information about those assets from the LDAP server.

We provide a built-in connectors for the most popular LDAP directory servers:

- Microsoft Active Directory
- Apache Directory Server (ApacheDS)
- Apple Open Directory
- Fedora Directory Server
- Novell eDirectory
- OpenDS
- OpenLDAP
- OpenLDAP Using Posix Schema
- Posix Schema for LDAP
- Sun Directory Server Enterprise Edition (DSEE)
- A generic LDAP directory server

LDAPS validation

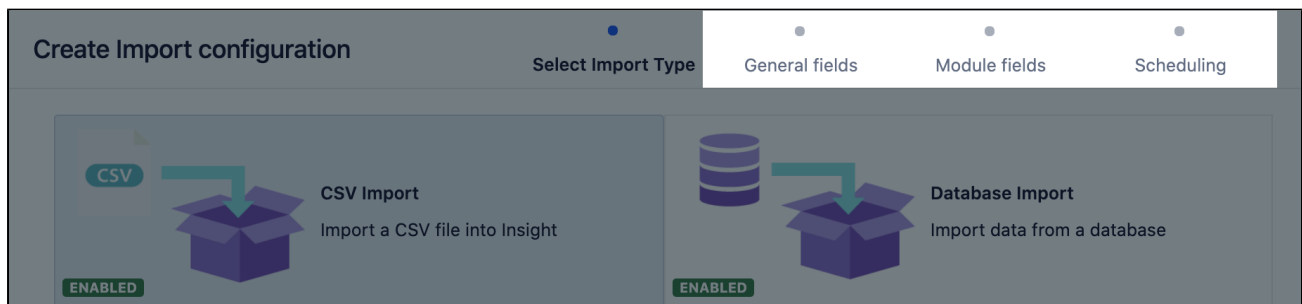
LDAPS (Secure LDAP) is supported and doesn't have any special requirements from Assets to work.

If you are trying to import from an LDAPS source, you can choose to validate the LDAP server certificate with an imported Certificate Authority (CA) certificate. If you select to validate the LDAP server certificate, you must import the root CA certificate from the CA that signed the LDAP server certificate, so your Jira can use the CA certificate to validate the LDAP server certificate. More information is explained [here](#).

Be sure to change the port to 3269. This is due to the fact that a GC (global catalog) server returns referrals on 389 which refers to the greaterAD "forest", but acts like a regular LDAP server on 3268 (and 3269 for LDAPS) when changing from LDAP to LDAPS.

Import fields

Once you've chosen your import type, you'll need to enter details about it. Here's the description of fields you should see in Assets.



Here are general fields, common for every import type:

Name	Description
Name	The name of the import.
Description	The description for your convenience.
Concatenator	<p>You can specify a default concatenator. When joining multiple data locators into one Insight attribute, this will be the default concatenator. One example could be to join two columns like "First name" and "Last name" into one attribute. So "Mathias" (first name) and "Edblom" (last name) will be concatenated as "Mathias Edblom" if using \s as concatenator.</p> <p>Enter \s for space-concatenated. To include a concatenate character, place the value between double quotes (i.e "\s").</p>
Empty Values	Defines what should happen when a Data Locator is empty. Should the import remove the attribute value or just ignores it and leave the current value as is.
Unknown Values	Defines what should happen if a Data Locator is unknown to Insight. This could happen with attribute types like "Status" and "Select". The value can be added as an option or just ignore the value.
Date Format	Format for date fields in import source to convert dates into Insight. If left empty, Insight will automatically try to find correct format. The format should be specified according to the Java SimpleDateFormat guidelines .
Date/Time Format	Format for date/time fields in import source to convert dates into Insight. If left empty, Insight will automatically try to find correct format. The format should be specified according to the Java SimpleDateFormat guidelines .

These fields are specific to an import type (module).

Setting	Description
URL	Protocol, Hostname and Port of the server running LDAP. Example: ldap://ldap.example.com:389
User DN	<p>The distinguished name of the user that the application will use when connecting to the directory server. Examples:</p> <ul style="list-style-type: none"> • <code>cn=administrator,cn=users,dc=ad,dc=example,dc=com</code> • <code>cn=user,dc=domain,dc=name</code> • <code>user@domain.name</code>
Password	The password of the user specified above.
Base DN	<p>The root distinguished name (DN) to use when running queries against the directory server. Examples:</p> <ul style="list-style-type: none"> • <code>o=example,c=com</code> • <code>cn=users,dc=ad,dc=example,dc=com</code> • For Microsoft Active Directory, specify the base DN in the following format:<code>dc=domain1,dc=local</code>. You will need to replace the <code>domain1</code> and <code>local</code> for your specific configuration. Microsoft Server provides a tool called <code>ldap.exe</code> which is useful for finding out and configuring the the LDAP structure of your server. <p>If you want specific Base DN in your object type see the Selector value below</p>

Search filter	<p>Defines the scoop of the filter search, default is(objectClass=*)which will give you all entries. If you only want Jira Users for example, you can set (objectClass=person). Note that the Users in LDAP need to have the the "objectClass" set to "person".</p> <p>The search filter is important in the way that it can affect the synchronization time.</p>
Search scope when importing	<p>Search scope determines how objects should be fetched from the LDAP. Default setting is ONE_LEVEL while the locators and structure are created with SUBTREE.</p>
Follow Referrals	<p>LDAP functionality to make sure you always get the correct data, even in a distributed LDAP environment.</p>
Include namespace	<p>This option is only applicable when creating an Assets object structure from an LDAP server. The option will append the namespace e.g. <i>cn=users,ou=company,dc=example,dc=com</i>to the object type description. The value is not used while performing synchronizations.</p>

⚠ Be sure you test the synchronization in a test environment before doing it in production.

Scheduling fields are responsible for keeping your data in sync:

Name	Description
Synchronizing Account	The Jira user to use when synchronize data into Insight
Cron Expression	The interval for the automatic synchronization.
Automatically Synchronize	If the import should be scheduled for automatic synchronization.

Pre-defined structure and configuration

In the next step, after you've filled in the required fields, Assets will ask you whether you want to create a predefined structure (object type mappings) and configuration (attribute mappings). Details of this will differ depending on the import type. Some object type mappings are disabled by default, so make sure to select the relevant ones.

i You can import users or groups from only one Organizational Unit (OU) during an Assets LDAP import. For more information, see [How to import users or groups from specific OUs with Assets LDAP import](#).

Here's some details for the LDAP import:

The structure will be created based on the result from the LDAP server. When creating the predefined structure a query will be sent to the LDAP server with the configuration specified and fetch the result. Based on the result an object type hierarchy will be created. Each node (identified by DN) that has children will be treated as an object type and created. The attributes belonging to the Assets object type will be the attributes found on the node in the LDAP server.

If the result returned by LDAP server retrieves objects that don't have children, then it will be not possible to create a predefined structure automatically and it should be created manually.

The predefined structure will create two additional attributes for each object type. The attribute CN (Common Name) will be used as label and the attribute DN (Distinguished Name) will be set with the property hidden.

All attributes created by the predefined structure in the LDAP import will be of type *Default Text*. If the data represent something else review the attributes and change them accordingly.

Example

Example LDAP structure	Resulting Assets Object Type Structure

The predefined configuration will query the LDAP server and create a configuration mapping based on the same criteria as the structure described above. As data locators all attributes found will be choosable with the addition of the CN (Common Name) and the DN (Distinguished Name).

The identifier will be set to DN for each object to uniquely identify each object from the LDAP server.

Since the predefined configuration will be different based on the connected LDAP server the following is one example mapping the Employees as seen in the previous example

Example

Employees			
Selector	ou=employees		
IQL	"DN" endswith "ou=employees,ou=riada,dc=nodomain"		
Missing objects	Ignore		
Empty Values	Use default		
Unknown Values	Use default		
Identifier	Data Locator	Insight Attribute	Object mapping (IQL)
<input type="checkbox"/>	DESCRIPTION	DESCRIPTION	
<input type="checkbox"/>	SEEALSO	SEEALSO	
<input type="checkbox"/>	SN	SN	
<input type="checkbox"/>	TELEPHONENUMBER	TELEPHONENUMBER	
<input type="checkbox"/>	OBJECTCLASS	OBJECTCLASS	
<input checked="" type="checkbox"/>	DN	DN	
<input type="checkbox"/>	CN	CN	

If the LDAP import is configured to import users one can use the REGEX configuration to split users in order to create multiple users.

Import configuration created

You can now view your import configuration, but it's not ready yet. You still need to create or review the object type and attribute mapping, and make sure there are no problems with your import configuration.

When you're ready, go to [2. Create object type and attribute mapping](#).

Before you go

In the next step, you'll create the object mapping settings. Here are some settings specific to the LDAP import type.

Object type mapping

Name	Description
Selector	<p>In the LDAP import type the Selector is prepended to the Base DN value before the search in LDAP is executed. The value is used to narrow down the structured tree in the LDAP to specific nodes.</p> <p>The search filter will be the same as specified in the general configuration but the selector will narrow the scope where the search filter is applied.</p> <p>For example:</p> <p>If the Base DN is <code>dc=ad,dc=example,dc=com</code> and the Selector is <code>cn=users</code> the resulting LDAP search base will be <code>cn=users,dc=ad,dc=example,dc=com</code>.</p>

Object schema import

Import structured data from an Assets object schema (Object Types, Attributes, and Objects). You can import data from another schema on the same server, or from an object schema on an entirely different server. [Learn more about importing](#)

Some use cases could be:

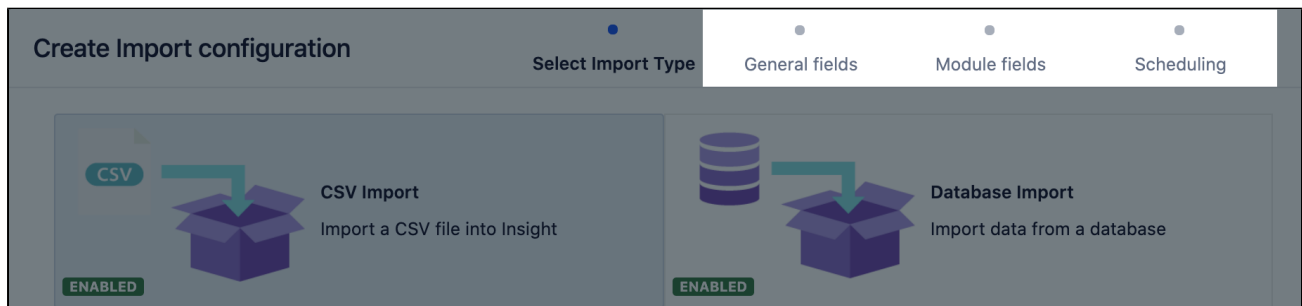
- Synchronize two different servers, each running a different object schema.
- Create a staging instance containing a subset of the full data from an object schema.
- Synchronize data between a master server and a set of child servers that contain a subset of the full data.

This import type won't include your schema's automation rules, roles, or any other configuration.

i You need to be an Assets Manager for an object schema to create, configure, and enable imports.

Import fields

Once you've chosen your import type, you'll need to enter details about it. Here's the description of fields you should see in Assets.



Here are general fields, common for every import type:

Name	Description
Name	The name of the import.
Description	The description for your convenience.
Concatenator	<p>You can specify a default concatenator. When joining multiple data locators into one Insight attribute, this will be the default concatenator. One example could be to join two columns like "First name" and "Last name" into one attribute. So "Mathias" (first name) and "Edblom" (last name) will be concatenated as "Mathias Edblom" if using \s as concatenator.</p> <p>Enter \s for space-concatenated. To include a concatenate character, place the value between double quotes (i.e "\s").</p>
Empty Values	Defines what should happen when a Data Locator is empty. Should the import remove the attribute value or just ignores it and leave the current value as is.
Unknown Values	Defines what should happen if a Data Locator is unknown to Insight. This could happen with attribute types like "Status" and "Select". The value can be added as an option or just ignore the value.
Date Format	Format for date fields in import source to convert dates into Insight. If left empty, Insight will automatically try to find correct format. The format should be specified according to the Java SimpleDateFormat guidelines .

Date/Time Format	Format for date/time fields in import source to convert dates into Insight. If left empty, Insight will automatically try to find correct format. The format should be specified according to the Java SimpleDateFormat guidelines .
-------------------------	--

These fields are specific to an import type (module).

Setting	Description
URL	The URL of the server that is running Jira with Assets.
Username	The username for authentication with the other server.
Password	The password for authentication with the other server.
Object Schema	The Name or Object Schema ID for the target server.

Scheduling fields are responsible for keeping your data in sync:

Name	Description
Synchronizing Account	The Jira user to use when synchronize data into Insight
Cron Expression	The interval for the automatic synchronization.
Automatically Synchronize	If the import should be scheduled for automatic synchronization.

Object collision

Because of the possibility the two synchronizing servers might contain objects with the same name, every object imported from another server is assigned an **External Hash Key** as an attribute. This hash key is built by appending the URL, schema ID, and the object key to create a longer, unique hash.

Pre-defined structure & configuration

In the next step, after you've filled in the required fields, Assets will ask you whether you want to create a predefined structure (object type mappings) and configuration (attribute mappings). Details of this will differ depending on the import type.

Here's some details for the object schema import:

- The predefined structure and configuration will be based upon the existing configuration of the target Assets object schema.
- Some object type mappings are disabled by default, so make sure to select the relevant ones.

Import configuration created

You can now view your import configuration, but it's not ready yet. You still need to map any object references or attributes between your source and Assets.

When you're ready, go to [2. Create object type and attribute mapping](#).

2. Create object type and attribute mapping

Once your base import configuration is created, you need to map your data to object types and their attributes. You might not need to create full mappings if you've chosen to create a predefined structure and configuration, but this page will help you to review and edit them, if needed.

View your import configuration

Just to make it easier to see where you are, here's a sample import configuration without mappings created (it's invalid because it doesn't have any object type mappings).



Create object type mapping

To create object type mapping:

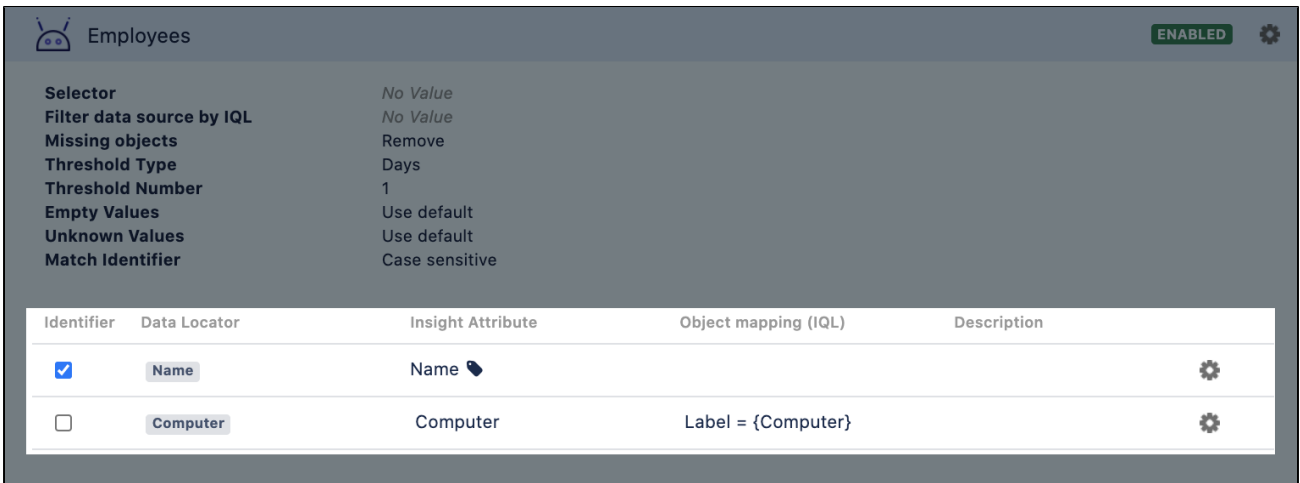
1. When viewing your import configuration, select **Create object type mapping**.
2. Specify the required fields.
3. Remember that every object type mapping imports your data into a single object type. If you have more object types, you'll create more object type mappings on the same import configuration. Before you start adding another object type mappings, complete the attribute mapping, as described below.

Field	Description
Object type	The Assets object type you want to import your data into. This value can't be changed, you need to create a new object type mapping to import into another object type.
Selector	This will differ based on your import type. See your type for details: <ul style="list-style-type: none">• CSV import (doesn't use the selector)• Database import• Jira users import• JSON import• LDAP import• Object schema import
Filter data source by AQL	<p>You can filter the data to be imported by AQL before it's actually imported. For example, if you're importing data into Computers object type, you can choose to only import PCs, and not Macs.</p> <p>Filter on object references</p> <p>You can't filter based on object references. These are handled separately after the object is created or updated.</p> <p>Here's a video that explain how to split your import source into different object types using AQL:</p>

<p>Missing objects</p>	<p>Choose what to do if an object is missing in the data source (but exists in Assets):</p> <ul style="list-style-type: none"> ▪ Update: Update the object with a predefined value to a specified attribute. The Threshold configuration will apply. ▪ Delete: Delete the object from Assets. The Threshold configuration will apply. ▪ Ignore: Just let the object be.
<p>Missing objects outbound references</p>	<p>Choose what to do if a referenced object (referenced from an existing object in your data source) is missing in the data source. For example, if an import is configured to import the referenced Host and Application object types, and one of them is missing.</p> <ul style="list-style-type: none"> ▪ Remove: The outbound reference to a missing object will be removed. ▪ Ignore: Leave the reference as is. <p>Assets will only consider data from the same import so if you have multiple import configurations, this functionality will not work.</p>
<p>Threshold type</p>	<p>When objects are missing in the data source, you can specify a threshold for the Update and Delete configurations. This means that Assets will wait with performing these actions until the conditions specified in the threshold are met (days or synchronizations).</p> <p>When you set the Threshold type to Synchronizations and the Threshold number to 1, missing objects will only be removed starting from the second synchronization. Similarly, if you set the Threshold number to 2, missing objects will be removed from the third synchronization onwards.</p>
<p>Threshold number</p>	<p>The number of days or synchronizations before performing the actions described above. If you want to remove missing objects from the first synchronization onwards, set the Threshold number to 0.</p>
<p>Empty values</p>	<p>Defines what to do if a Data locator is empty. Should the import remove the attribute value or just ignore it and leave the current value?</p> <p>This overrides the 'General fields' configuration of your import type for this object mapping.</p>
<p>Unknown values</p>	<p>Defines what to do if a Data locator is unknown to Assets. This can happen with attribute types like "Status" and "Select". The value can be added as an option or just ignored.</p> <p>This overrides the General fields configuration of your import type for this object mapping.</p>
<p>Match identifier</p>	<p>Case sensitive - Identifier AQL will be built up with STRICT_EQUAL (==) which means that for text attributes, identifiers will be compared respecting case, 'Data Value' will NOT be considered the same as 'data value'.</p> <p>Case insensitive - Identifier AQL will be built up with EQUAL (=) which means that for text attributes, identifiers will be compared ignoring case, 'Data Value' will be considered the same as 'data value'.</p>

Create attribute mapping

Once your object type mapping is created, you can expand it to see the attribute mapping. Attribute mapping is created for each object type mapping separately, as you'll map data locators from your data source to the object type's attributes in Assets. There needs to be at least one attribute mapping for every object type mapping.



Specifies if the attribute should be considered a part of the object identifier. If multiple attributes are used as identifiers, the combined value will be used to identify the object and duplicates. This is used to construct an internal AQL responsible for identifying objects.

For example, if two import attributes are configured for Assets attributes First Name and Last Name, and specified as identifiers, the following AQL will be used to identify objects in Assets:

```
"First Name" == <first name value from source> AND "Last Name" == <last name value from source>
```

All attributes can be identifiers. But, for best performance, try to use the default attribute types as much as you can. For example, if using "User" attribute as identifier, we will have to query the Jira API to find users from the data source for the configured data locator. This means that Jira might be a bottle neck in this case if you have a large User directory. Try in this case to add another attribute for the user key (Default / Text) and the import will be much faster.

If you are importing objects from a CSV file into Assets, you must identify the objects by the attribute that is assigned as a label, not their name or key. The object key is generated automatically when an object is created manually or via import. Even if the object key is present in the source file (say CSV or JSON), a new object and its key will be generated instead of using the one in the source file, especially if the object key is the identifier. This could result in multiple duplicates.

Every import type needs to provide data locators. It's what Assets recognizes as a category that holds your data. For example, in CSV import, all columns are data locators.

The Assets attribute to which you want to map the data locator, and import its data.

Only used for Assets attributes of type Object (references between objects).

In this case, you need to provide an AQL to fetch the objects to set, depending on the data provided by the Data Locator. In the examples that follows, *Name* is a reference to the value in Assets that the referenced object has, in the attribute *Name*.

You can use place holders in the AQL like:

- For single attributes (max cardinality equals 1) the AQL with the placeholder could look like this:

```
Name = ${DATA LOCATOR}
```

- For multiple attributes (max cardinality larger then 1) the AQL with the placeholder could look like this:

```
Name IN (${DATA LOCATOR${0}})
```

The "0" (zero) stands for all values, and 1 for the first value, 2 for the second value and so on.

For more info, see [Understanding import concepts](#).

Description for the attribute mapping, just for convenience.

You can access other configuration by selecting the cog icon next to your attribute mapping.

Concatenator

When joining multiple data locators into one Assets attribute, this works as the default concatenator.

Regular expression

You can use regular expressions to edit the value from the import source.

For the regular expression we use Java syntax <https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>. If the regular expression does not find any match the attribute is left empty. It is possible to extract from the value given by the Data Locator by making what you wish to extract the first group in the regular expression, for example using the regular expression "(w*)\W" (quotation marks excluded) on the value "Blue Whale" yields "Blue" to the corresponding attribute. If there is no group present and the regex finds a match the whole value is given.

Some examples:

Purpose	Pattern	Example
Validate that input contains at least one digit	\d	"foo 123 bar" is returned as is but "foo bar" gives empty attribute value
Extract first digit	(\d)	"foo 123 bar" gives 1 but "foobar" gives empty attribute value
Extract only first word of text	(\w*)\W	"Blue whale" becomes "Blue"
Shorten number to show only two decimals	(\d*\.\d\d)	12.3456 becomes 12.34

Next steps

When you're ready with your mappings, go to [3. Inspect your import configuration](#).

3. Inspect your import configuration

You can quickly view any elements of your import configuration that have been changed or are missing (from the expected configuration). Although you should strive for perfection, any problems here doesn't necessarily mean that your configuration is wrong, if you mapped things differently on purpose.

To inspect your import configuration,

1. Open your import configuration.
2. Select the cog icon next to it, and select **Inspect configuration**.

Results

Here are some sample results:

- **Changed:**Data locator is called differently in your data source and the attribute it's mapped to. This doesn't break your import (if the mapping is done properly), but you might want to have a look why you have these differences.
- **Missing:**The expected attribute mapping is missing.

If you're seeing problems, go back to your import configuration and check what's wrong. Make sure the data you're importing is mapped to the relevant attributes.

Next steps

When your import configuration looks fine, go to [4. Enable and sync your import](#).

4. Enable and sync your import

When your import configuration is ready (you're just about starting to import your data), you can enable your mappings and the configuration itself, and sync them.

Enabling object type mappings

To enable the object mappings:

1. Select the cog icon next to the object type mapping, and select **Enable**.
2. You can keep some mappings disabled if you're not ready to import the related data.
3. You can also synchronize only some of your object type mappings. In any case, you need to enable your import configuration before you can sync any mappings.

Enabling and syncing your import

To enable and sync your import configuration:

1. Select the cog icon next to the object type mapping, and select **Enable**.
2. If your mappings are ready, also select **Synchronize** for your whole configuration. This will sync any object type mappings that are enabled.
3. To configure regular synchronization for your import (if you haven't done that when creating your import), go to **Configure**, and move to **Scheduling fields**.

Finishing up

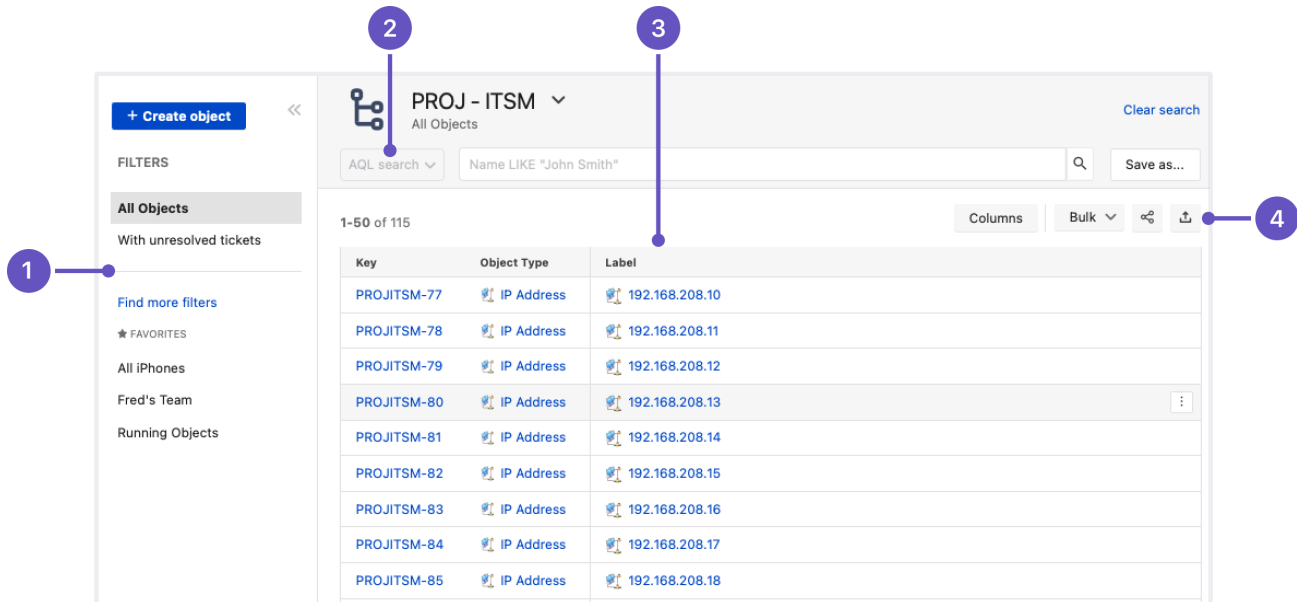
If everything went ok, your data should be imported to your object types. If you're missing something, go back to your configuration and edit the fields or the mappings. It might be useful to watch some of our videos on importing to get a better understanding of how imports work. [Learn more about importing concepts](#)

Searching for objects

To open the object view search, select **Assets > Search for objects**. You can either AQL or FreeText in the text box:

- **FreeText** will search the text of all objects and object attributes for relevant strings, but will not search any non-string data.
- **AQL** can be used to create more advanced queries using logical statements, placeholders, and references. In the object search screen, the built-in autocomplete will assist you in creating AQL searches. For more info, see [AQL - Assets Query Language](#).

When you launch any of your favorite search filters, you get a smoother search feedback when the results are being displayed.



1. **Filters:** List of saved and favorite filters, just like in issue search.
2. **Type of search:** Here you can switch between basic (FreeText) and advanced (AQL) search.
3. **Objects:** Search results, or the list of objects found.
4. **Bulk actions:** Actions that you can perform on your search results.

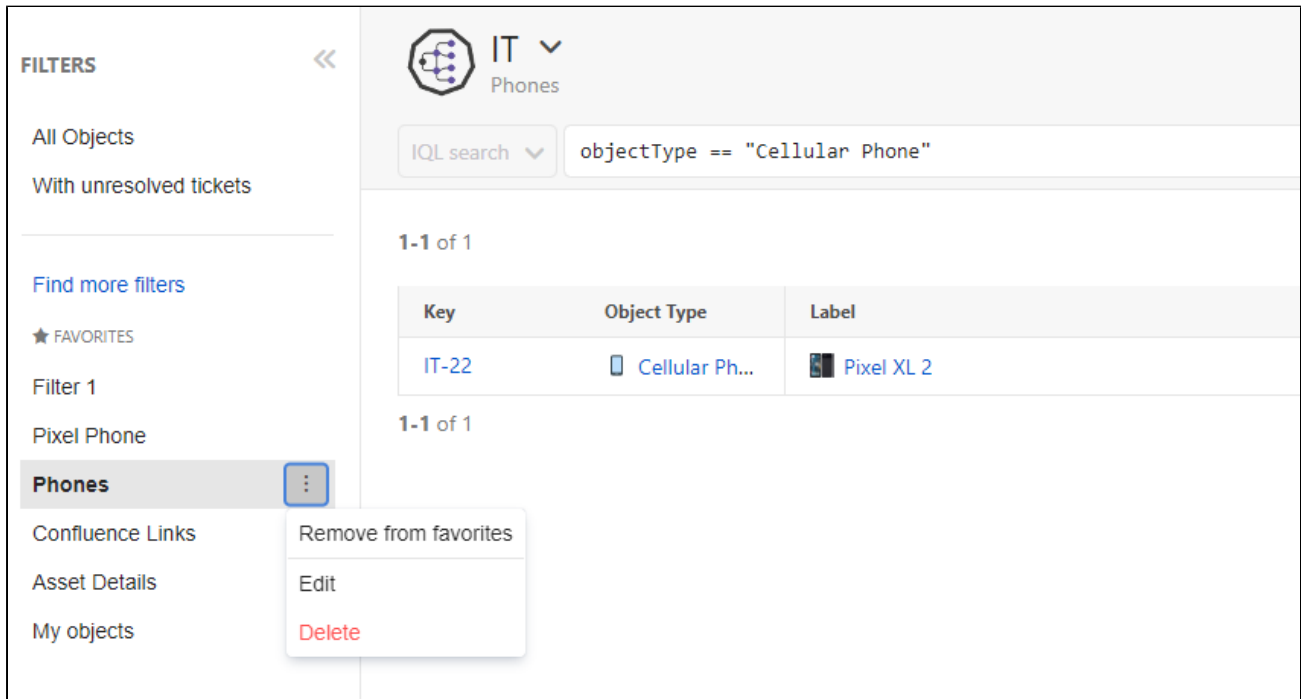
Manage your search results

Here's what you can do with your search results:

Action	Description
Rearrange columns	Drag and drop columns to rearrange them. Use the Columns option to show or hide specific columns, and pin them so they're permanently displayed.
View object's graph	Create a new object from your search by selecting Create object in the panel at top-left.
Edit, move, or delete objects	Edit, move or delete objects that are the results of your search using the options button that appears on the right when you hover over your results.
Share or export search results	When viewing your search results, just select Share or Export to share your results or export them into CSV.
Save search results	Save your search results for later use by selecting Save as at the top-right. Saved search results become filters, and are displayed in the left-hand side menu.

Manage your filters

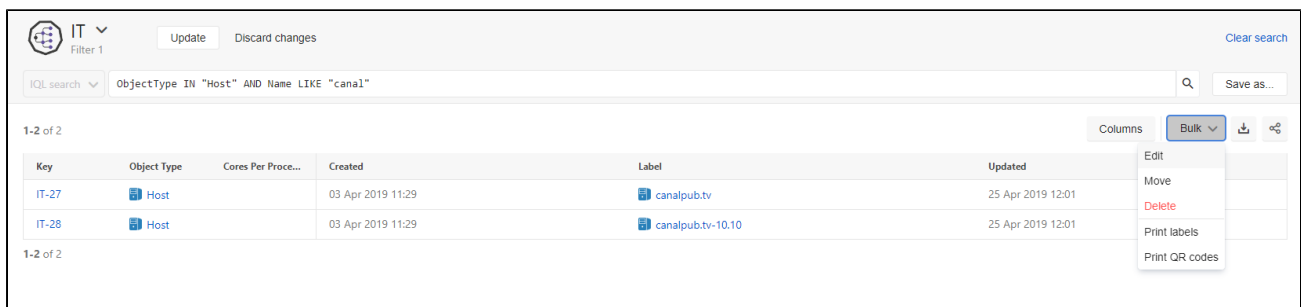
Searches that you saved, or filters that have been shared with you, appear as filters in the panel on the left. You can select the **Options** menu next to a filter name to edit, delete, or remove it from your favorites.



Modify multiple objects with bulk actions

You can also modify multiple objects from your search results.

1. When viewing your search results, select **Bulk**, and choose one of the bulk actions.
2. In the next step, you'll be able to select specific objects by selecting them. This helps you to further limit the objects you want to complete a bulk action on.



Advanced search: AQL - Assets Query Language

Ready for some more advanced information? AQL can be used to create advanced queries using logical statements, placeholders, and references. You'll see it not only to search for your objects, but also to call them out when configuring custom fields and relations between objects.

- [Advanced searching: AQL - Assets Query Language](#)
- [Advanced: Placeholders](#)
- [Assets AQL functions](#)

Advanced searching: AQL - Assets Query Language

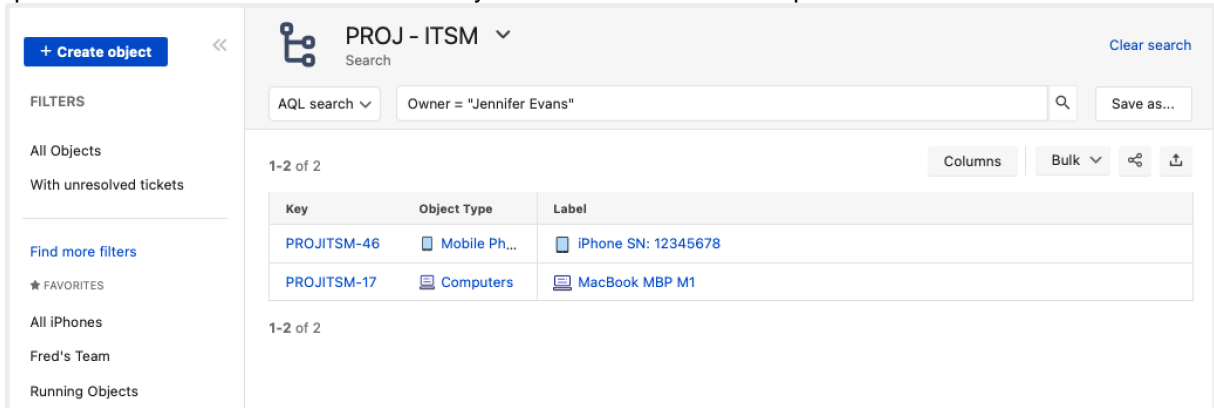
AQL (Assets Query Language) is a language format used in Assets to create search queries for one or more objects. Using AQL, you can return any object or group of objects in Assets in a search, filter objects, modify objects, create custom fields, automations and post-functions, and more.

Basic syntax

The basic syntax of an AQL query is **<attribute> <operator> <value/function>**. One or more objects is returned by the query when the attributes of these objects match the operator and value specified.

Example

- A basic AQL query that returns all objects for which the Owner is "Jennifer Evans". Note the quotations around "Jennifer Evans" they're needed because of a space in the value name.



The screenshot shows the Assets search interface for 'PROJ - ITSM'. The search bar contains the query 'Owner = "Jennifer Evans"'. The results table shows two objects:

Key	Object Type	Label
PROJITSM-46	Mobile Ph...	iPhone SN: 12345678
PROJITSM-17	Computers	MacBook MBP M1

Syntax for special characters

AQL has a defined syntax and must be entered exactly.

- AQL is case-sensitive.
- If you are using an expression where the value or attribute contains a space, you must include quotations surrounding the value, for example "Ted Anderson".
- If you are using an expression where the value or attribute contains quotation marks, you must escape the quotes by surrounding them with backslashes. For example, if you have an object name such as *15" Screen*, to search for it enter: "15\" Screen"
- The attribute name that you specify in the AQL must exist in your Assets schema. If not, the AQL will be considered invalid. The attribute name is case sensitive.

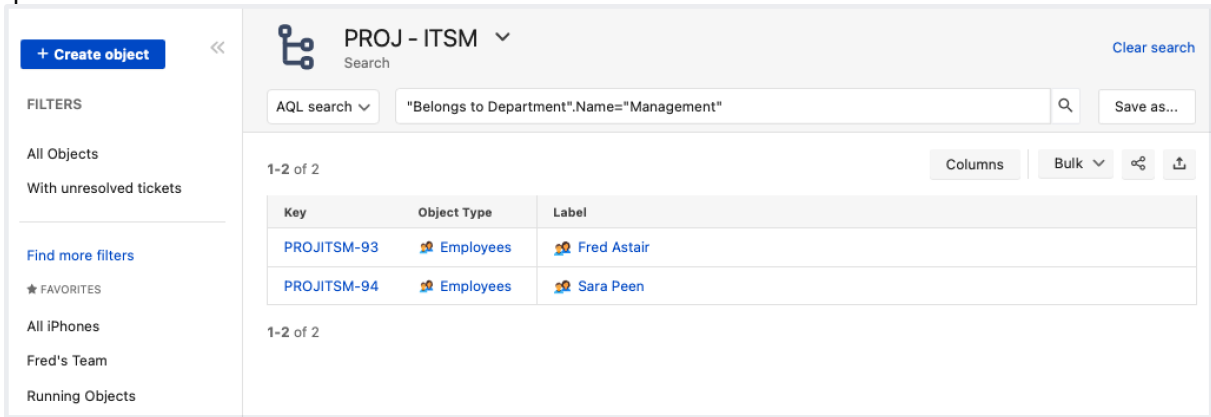
Dot notation

Dot notation is used in AQL to travel down a reference chain of objects. The format **<attribute>.<attribute> <operator> <value/function>** will return information based upon objects referenced by the parent object.

Example

- In this case, the Employee object type has a referenced attribute called "Belongs to Department". The query returns all the Employees which belong to the "Management" department.

- Note that since the referenced attribute contains spaces, it has been enclosed with a pair of double quotes.



Keywords

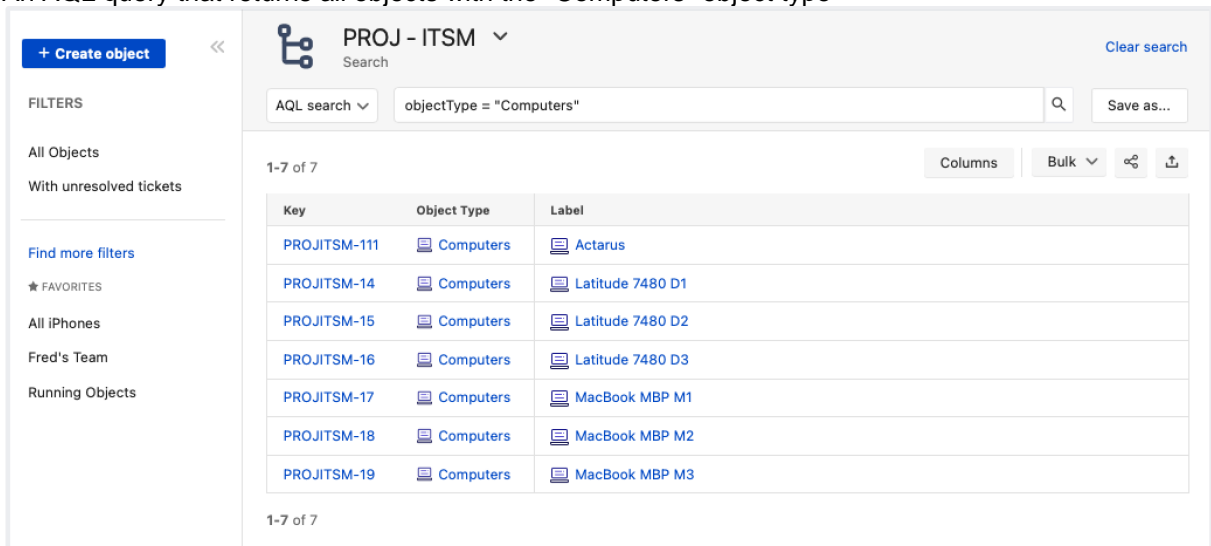
You can use keywords in AQL to return one or more objects based upon properties of those objects instead of the object's attributes. The syntax looks like this: **<keyword><operator> <value/function>**.

For example, you could return all objects of a specific object type using the **objectType** keyword, or all objects of any type with a certain attribute by using the **anyAttribute** keyword. The table below describes the keywords supported along with their respective examples.

Keyword	Description
objectType	You can limit the search result to a specific object type name, e.g. objectType = "Employment Start Date". Note that the since the attribute contains spaces, it is enclosed within a pair of double quotes.
objectTypeid	You can limit the search result on object type ids. e.g. objectTypeid in (1, 2).
anyAttribute	You can search all attributes on all objects for a relevant match. e.g. "anyAttribute = 123.123.123.123". <i>Note that the use of this keyword may cause delays in searching results. The larger the Assets installation, the more time it will take to execute a query with this keyword.</i>
object	You can limit the search to the object, e.g. "object having inboundReferences()" will search all objects having any inbound references to it.
objectId	You can find an object by object Id, e.g. "objectId = 114". Note that the object id is the number from the Key of the object, but without the prefix. E.g. if the Key of your object is ITSM-1111, then the prefix is ITSM and the object id is 1111. Note that the Key could change when you move objects across schemas.

Example

- An AQL query that returns all objects with the "Computers" object type



Operators

Operators allow you to create more detailed logical expressions.

The table describes the operators supported in AQL.

Operator	Description	Example AQL query
=	Equality test for case insensitive values.	"Office = Stockholm" Checks if the Office attribute has a value equal to Stockholm or STOCKHOLM
==	Equality test for case sensitive values.	"Office==Stockholm" Checks if the Office attribute has a value equal to Stockholm considering the case of the input provided.
!=	Inequality test	objecttype=Employee and Office!=Stockholm Checks if the Employee object has an attribute Office whose value is NOT equal to Stockholm.
<	Less than test.	"Price < 2000" Checks if the Price is less than 2000 dollars.
>	Greater than test.	"Price > 2000" Checks if the Price is greater than 2000 dollars.
<=	Less than or equal to test	"Price <= 2000" Checks if the Price is less than or equal to 2000 dollars.
>=	Greater than or equal to test.	"Price >= 2000" Checks if the Price is greater than or equal to 2000 dollars.
like	Matches a value with any subset of input in the query. It is case insensitive.	"objecttype=Employees and Office like Stock"

		Returns all objects of <i>Employees</i> type which have an <i>Office</i> attribute value that contains the characters 'Stock' or 'STOCK'
not like	Excludes values which match with any subset of input in the query.	<i>"objecttype=Employees and Office not like Stock"</i> Returns all objects of <i>Employees</i> type which have an <i>Office</i> attribute value that DO NOT contain the characters 'Stock'
in()	Finds a match in the given arguments and returns results.	<i>Office in ("Stockholm", "Oslo", "San Jose")</i> Returns all objects of <i>Officetype</i> which HAVE one of these values: <i>Stockholm, Oslo, San Jose</i> .
not in()	Excludes the results for which a match is found in the given arguments.	<i>Office not in ("Stockholm", "Oslo", "San Jose")</i> Returns all objects of <i>Officetype</i> which DO NOT HAVE one of these values: <i>Stockholm, Oslo, San Jose</i> .
startswith	Finds a match whose value starts with the given input. It is case insensitive.	<i>"Office startsWith St"</i> Returns results which match values of <i>Officetype</i> starting with the characters "St" or "ST"
endswith	Finds a match whose value ends with the given input. It is case insensitive.	<i>"Office endsWith St"</i> Returns results which match values of <i>Officetype</i> ending with the characters "St" or "ST"
is	Helps test whether a value exists or not.	<i>"Office is EMPTY"</i> Checks whether the value of the <i>Officetype</i> exists and returns results accordingly. <i>"Office is not EMPTY"</i> Checks whether the value of the <i>Officetype</i> is not empty.
dot operator (.)	Helps navigate the <i>Referenced</i> types attributes for an object. This operator is commonly used in: <i>inboundReferences()</i> or <i>inR()</i> functions. <i>outboundReferences()</i> or <i>outR()</i> functions. Order by clause.	<i>"Country.Office = Stockholm"</i> The dot operator here navigates to the referenced object <i>Office</i> in the attribute <i>Country</i> and compares <i>Office</i> with the value <i>Stockholm</i> .
having	Used with either the <i>inboundReferences()</i> OR <i>outboundReferences()</i> functions	<i>"object having inboundReferences()"</i> Returns all objects having inbound references.
not having	Used with either the <i>inboundReferences()</i> OR <i>outboundReferences()</i> functions	<i>"object not having inboundReferences()"</i> Excludes all objects having inbound references and returns results.

Example

- An AQL query that uses the **IN** operator and **objectType** to return objects of three different object types.

The screenshot shows the Jira Service Management interface for a project named 'PROJ - ITSM'. The AQL search bar contains the query: `objectType IN ("Computers", "Employees", "CPU")`. The results table displays 21 objects, categorized by their object type: Computers, Employees, and CPU.

Key	Object Type	Label
PROJITSM-111	Computers	Actarus
PROJITSM-92	Employees	Anton Dry
PROJITSM-93	Employees	Fred Astair
PROJITSM-1	CPU	Intel(R) Xeon(R) CPU E7- 4830 @ 2.13GHz
PROJITSM-2	CPU	Intel(R) Xeon(R) CPU E7- 4830 @ 2.13GHz
PROJITSM-3	CPU	Intel(R) Xeon(R) CPU E7- 4830 @ 2.13GHz
PROJITSM-4	CPU	Intel(R) Xeon(R) CPU E7- 4830 @ 2.13GHz
PROJITSM-5	CPU	Intel(R) Xeon(R) CPU E7- 4830 @ 2.13GHz
PROJITSM-6	CPU	Intel(R) Xeon(R) CPU E7- 4830 @ 2.13GHz
PROJITSM-7	CPU	Intel(R) Xeon(R) CPU E7- 4830 @ 2.13GHz
PROJITSM-8	CPU	Intel(R) Xeon(R) CPU E7- 4830 @ 2.13GHz
PROJITSM-14	Computers	Latitude 7480 D1
PROJITSM-15	Computers	Latitude 7480 D2
PROJITSM-16	Computers	Latitude 7480 D3
PROJITSM-17	Computers	MacBook MBP M1
PROJITSM-18	Computers	MacBook MBP M2
PROJITSM-19	Computers	MacBook MBP M3
PROJITSM-94	Employees	Sara Peen
PROJITSM-95	Employees	Tess Aguila
PROJITSM-112	CPU	Virtual CPU
PROJITSM-96	Employees	Wes Shults

Combination operators

You may use operators such as AND/OR to create larger and more complex AQL expressions.

Example

- An AQL query that includes two statements linked with the AND operator

The screenshot shows the Jira Service Management interface for a project named 'PROJ - ITSM'. The AQL search bar contains the query: `objectType = "Computers" AND "Operating System" = "Microsoft Windows 10 Pro"`. The results table displays 2 objects, both of type 'Computers'.

Key	Object Type	Label
PROJITSM-111	Computers	Actarus
PROJITSM-14	Computers	Latitude 7480 D1

Functions

You can use different functions to supply dynamic values to AQL expressions.

Type	Function name	Description
Date	now() startOfDay() endOfDay()	You can use a range of functions to write queries which involve date and time. We use m for minutes, h for hours, d for days and w for weeks to represent relative time.
	startOfWeek() endOfWeek() startOfMonth() endOfMonth() startOfYear() endOfYear()	e.g. A query with a condition like: <i>Created > "now(-2h 15m)"</i> returns all objects created in the last 2 hours and 15 minutes. e.g. A query containing something like: <i>objectType = Employees and "Employment End Date" < endOfMonth(-90d)</i> returns all Employee objects whose <i>Employment End Date</i> falls before 90 days from the current month's end date. e.g. You may also check for an upcoming date, e.g. check when the license of a software expires by the end of the year. Your query can then include something like <i>:licenseEndDate = endOfYear()</i> You can use all other date functions in a similar manner.
IP Address	CIDR(IP RANGE)	CIDR(IP RANGE)-Filter on IP ranges e.g. <i>"IP Address" IN CIDR("192.0.0.0/8")</i> <i>"IP Address" IN CIDR("192.168.0.0/16")</i>
User	currentUser()	You can filter on user attributes connected to the current (logged in) user by invoking this function in your AQL query. Note that the attribute used in the query for filtering needs to be of type User. e.g. <i>objecttype = Computer and User = currentUser()</i> This function will work when a currentUser is selected, ie. the user is logged in.
	currentReporter()	You can filter on user attributes connected to current reporter in custom fields by invoking this function in your AQL query. Note that the attribute used in the query for filtering needs to be of type User. e.g. <i>User = currentReporter()</i> This function will only work when an issue is selected, and refers to the reporter that appears in the current issue.
	user (user1, user2, ..)	You can filter on objects which have a reference to the users that you provide in the argument list of the function. The attribute used to filter must be of User type. This function will work with multiple arguments only if the User type attribute that you filter on allows multiple values i.e, the cardinality for it is more than one. e.g. An object type <i>Team</i> has an attribute <i>Member</i> . This attribute is of User type. Additionally, this attribute has been configured to have a cardinality of 3. If you want to search a set of <i>Team</i> objects where the users: <i>admin</i> and <i>manager</i> are its members, you can write the following query: <i>objecttype=Team and Member having user("admin", "manager")</i>

Group	group (group1, group2,...)	You can filter on any object connected to a user within a specific group. The attribute used to filter has to be of <i>Usertype</i> . e.g. <i>User in group("jira-users", "jira-administrators")</i>
	user (user1, user2, ...)	Filter on any object connected to a user within a specific group. The attribute used to filter has to be of <i>GroupType</i> . e.g. <i>Group having user("currentReporter()")</i>
Project	currentProject()	Filter on any object connected to the currently selected Jira project. Works only in the context of a ticket. e.g. <i>Project = currentProject()</i>

Reference functions

Reference functions are functions that take two arguments - AQL and/or a reference-type argument. Essentially, you can use reference functions to run an AQL query on a subset of objects of a particular reference type. Such a query will run on a small subset of the total objects, which allows you to limit results and/or processing time.

- The **AQL argument** can be an arbitrary AQL including an AQL with reference function.
- The **Reference Type argument** is optional.

Sr. no	Name	Description
a	<ul style="list-style-type: none"> • inboundReferences (AQL) • inR (AQL) 	<p>Filter objects having inbound references where the referenced objects match the AQL query p</p> <p>e.g. An AQL query like: <code>object having inboundReferences()</code> will return all objects having inbound</p> <p>But an example query like: <code>object having inboundReferences(Name="John")</code> will return all objec</p>
b	<ul style="list-style-type: none"> • inboundReferences (AQL, referenceTypes) • inR (AQL, 	<p>This is a variant of the <code>inboundReferences(AQL)</code> function described in (a)</p> <p>Using this, you can filter the inbound referenced objects further by providing the Reference Ty</p> <p>Reference Type is the Additional Value field that you provide on an attribute when you define</p>

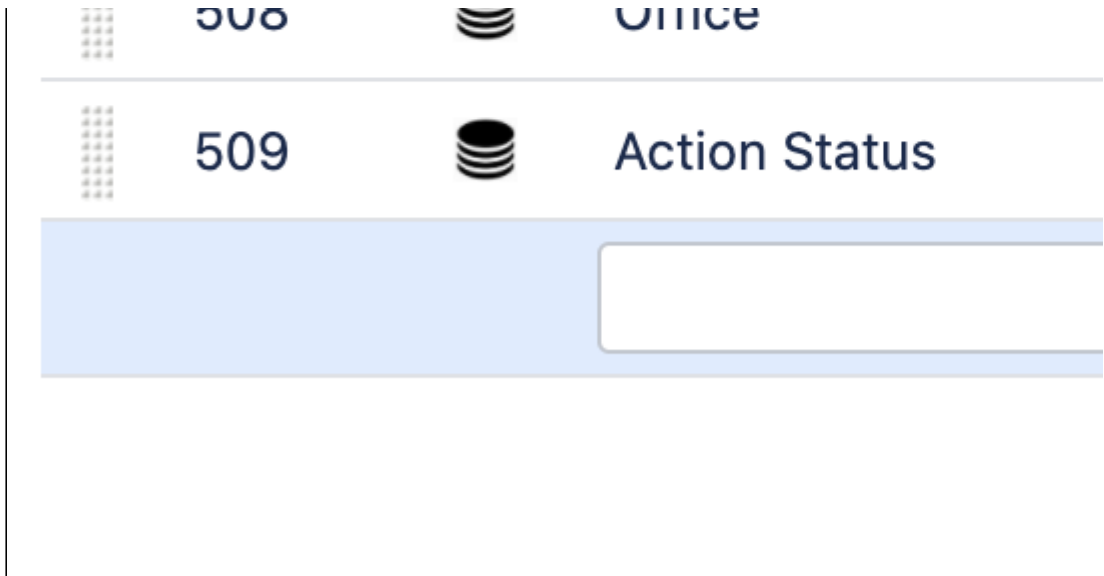
ref
Type
s)



Employees

No description

	Id		Name
	313		Key
	314		Name
	315		Created
	316		Updated
	501		Active
	502		Email Address
	503		Username
	504		Manager
	505		Phone
	506		Computer
	507		Country
	508		Office



e.g. An AQL query like this: *object having inR(objectType = "File System", refType IN ("Deper* will return objects which have inboundreferencedobjects of "File System" and only for thoseFi
Similarly, an AQL query like: *object having inR(objectType = "File System", refType IN ("Depe* will return objects which have inboundreferencedobjects ofFile Systemand for thoseFile Syste

c

- out
bo
un
dR
efe
ren
ces
(A
QL)
- out
R
(A
QL)

Filter objects having outbound references where the referenced objects match the AQL query
e.g. An AQL query like: *object having outboundReferences()* will return all objects havingoutb
But an example query like: *object having outboundReferences(Name="John")* will returnall obje

d

- out
bo
un
dR
efe
ren
ces
(A
QL
,
ref
ere
nc
eT
yp
es)
- out
R
(A
QL)

This is a variant of the outboundReferences(AQL) function described in (b)
Using this, you can filter the outbound referenced objects further by providing the**Reference** 1
Reference Type is the**Additional Value**field that you provide on an attribute when you define
e.g. An AQL query like this: *object having outR(objectType = "Employees", refType IN ("Locat*
will return objects which haveoutboundreferencedobjects ofEmployeesand only for thoseEmp
Similarly, an AQL query like: *object having outR(objectType = "Employees", refType IN ("Loca*
will return objects which haveoutbound referencedobjects ofEmployeesand for thoseEmploye

	, ref Type s)
--	---------------

Example

- An AQL query that uses the inR function to return all objects that are of object type "Computers", and have inbound references of type "A reference" or "IP Address"

PROJ - ITSM Search

AQL search: object having inR(objectType="Computers", refType IN ("A reference", "IP Address"))

1-10 of 10

Key	Object Type	Label
PROJITSM-77	IP Address	192.168.208.10
PROJITSM-78	IP Address	192.168.208.11
PROJITSM-79	IP Address	192.168.208.12
PROJITSM-80	IP Address	192.168.208.13
PROJITSM-82	IP Address	192.168.208.15
PROJITSM-83	IP Address	192.168.208.16
PROJITSM-87	IP Address	192.168.208.2
PROJITSM-2	CPU	Intel(R) Xeon(R) CPU E7- 4830 @ 2.13GHz
PROJITSM-5	CPU	Intel(R) Xeon(R) CPU E7- 4830 @ 2.13GHz
PROJITSM-112	CPU	Virtual CPU

1-10 of 10

Using the connectedTickets() function

The **connectedTickets()** function is used to filter objects having tickets connected to them. Specific Jira issues may be selected by providing a proper Jira Query Language (JQL) query. If no JQL query is provided, all objects having Jira issues connected are returned.

Name	Description
connectedTickets()	All objects having tickets connected to them are returned.
connectedTickets(JQL query)	Object having tickets connected to them that match given JQL query are returned.

Example

- This query runs a JQL query (labels IS empty) on all connected issues, and then returns objects based on the results

The screenshot shows a search interface for 'PROJ - ITSM'. The search query is 'object HAVING connectedTickets(labels is empty)'. The results are displayed in a table with 9 rows. The table has three columns: Key, Object Type, and Label.

Key	Object Type	Label
PROJITSM-97	Memory	32GB DDR4-2400 ECC REG DIMM
PROJITSM-34	Network I...	Apc Network Management Card 2
PROJITSM-72	Business ...	Billing Service
PROJITSM-73	Business ...	Email Service
PROJITSM-74	Business ...	ERP Service
PROJITSM-26	Servers	exchange.prod.app1
PROJITSM-30	Servers	oracledb-11g-prod
PROJITSM-75	Business ...	Payroll Service
PROJITSM-31	Servers	sap-erp-prod

Using the objectTypeAndChildren() function

This function is used to return objects (and their children) of a specific object type.

Name	Description
objectTypeAndChildren (Name)	Filter objects based on the object type specified by theNameand its children. If the name contains spaces, make sure you enclose it within a pair of double quotes.
objectTypeAndChildren (ID)	Filter objects based on the object type specified by the IDand its children.

Example

- An AQL query that returns all objects and child objects from the "Hardware" object type

PROJ - ITSM Search

AQL search

1-50 of 53

Key	Object Type	Label
PROJITSM-97	Memory	32GB DDR4-2400 ECC REG DIMM
PROJITSM-98	Memory	32GB DDR4-2400 ECC REG DIMM
PROJITSM-99	Memory	32GB DDR4-2400 ECC REG DIMM
PROJITSM-100	Memory	32GB DDR4-2400 ECC REG DIMM
PROJITSM-101	Memory	32GB DDR4-2400 ECC REG DIMM
PROJITSM-102	Memory	32GB DDR4-2400 ECC REG DIMM
PROJITSM-103	Memory	32GB DDR4-2400 ECC REG DIMM
PROJITSM-104	Memory	32GB DDR4-2400 ECC REG DIMM
PROJITSM-105	Memory	32GB DDR4-2400 ECC REG DIMM
PROJITSM-106	Memory	32GB DDR4-2400 ECC REG DIMM
PROJITSM-107	Memory	32GB DDR4-2400 ECC REG DIMM
PROJITSM-108	Memory	32GB DDR4-2400 ECC REG DIMM
PROJITSM-109	Memory	32GB DDR4-2400 ECC REG DIMM
PROJITSM-110	Memory	32GB DDR4-2400 ECC REG DIMM
PROJITSM-111	Computers	Actarus
PROJITSM-34	Network I...	Apc Network Management Card 2
PROJITSM-35	Network I...	Apc Network Management Card 2

Functions, references, and AQL can be combined in powerful ways. For example, you could add multiple object references to a custom field attached to an object, and then search those references for a specific key:

```
object HAVING inboundReferences(Key IN (${MyCustomField${0}}))
```

Or a specific label, by using dot notation:

```
object HAVING inboundReferences(Label IN (${Portfolios.label${0}}))
```

Note that the above two queries make use of Assets placeholders.

Ordering

You can order the results of your query by adding the following suffix to any AQL:

```
order by [AttributeName|label] [asc|desc]
```

Good to know

- If you do not specify an order by clause, the default will be ascending order by the *label* attribute of an object type.
- If the attribute specified in the order by clause is of the object reference type you can use dot notation to order by attributes on the referenced object. E.g. if you want to order by the referenced *Department* object of an *Employee* object, you can mention the clause like: `order by Employee.Department`. This can

be done in unlimited depth. However, note that the every dot in the order by clause will decrease the performance of that particular AQL.

- Missing values will appear at the top of the list. This will hold true if the order in the query is ascending order : "asc".
- The attribute name specified in the AQL must exist in Assets. If not, the AQL will be considered invalid. The attribute name is case sensitive.
- If the results **do not** contain the attribute specified in the "order by" clause the order of the objects returned will be arbitrary.
- The placeholder label can be used instead of the attribute name to order the objects by their configured label.

Example

- These search results are ordered in descending order by their keys

The screenshot shows a search interface for 'PROJ - ITSM'. The search query is 'objectType IN ("Computers") order by Key DESC'. The results are displayed in a table with 7 rows, ordered by key in descending order. The table has columns for 'Key', 'Object Type', and 'Label'.

Key	Object Type	Label
PROJITSM-111	Computers	Actarus
PROJITSM-19	Computers	MacBook MBP M3
PROJITSM-18	Computers	MacBook MBP M2
PROJITSM-17	Computers	MacBook MBP M1
PROJITSM-16	Computers	Latitude 7480 D3
PROJITSM-15	Computers	Latitude 7480 D2
PROJITSM-14	Computers	Latitude 7480 D1

Advanced: Placeholders

Assets supports a wide range of different placeholders that allow you to dynamically replace a specified placeholder with anything from an Issue field value, Assets Object attribute value or other Assets related information depending on the context.

When to use placeholders?

Here are some use cases that will show you how placeholders work:

- Get an email when a license is about to expire, with the **name** and **end date** of that specific license.
- Set/update an object attribute with the **value** from an issue field when issue is transitioned to a specific status.

What these use cases have in common is that you know where to find the value, but not what the value will be at the time when you'll need it. Placeholders will find the right value.

Placeholder structure

The placeholders are structured like this:

```

${Name of placeholder}
or
${Name of placeholder${Number}}

```

The following table describes how placeholders are used, with some examples:

Placeholder	What it does	Example placeholder	Replaced with value
\${Name of placeholder}	Replaces the content <code>\${Name of placeholder}</code> with the value matched by <i>Name of placeholder</i>	<code>\${Name}</code>	One object
		<code>\${Email}</code>	myemail@fakse.se , anotheremail@fakse.se
\${Name of placeholder\${0}}	Replaces the content <code>\${Name of placeholder\${0}}</code> with all values matched by <i>Name of placeholder</i> . This is used to match multiple values e.g. when populating an attribute with multiple cardinality and all values will be one instance of that attribute. When replaced as text this is similar to the above example but when replaced in a multi value context this notation will render multiple values but the above will render one value as a comma separated string.	<code>\${Price\${0}}</code>	12
		<code>\${Email\${0}}</code>	myemail@fakse.se , anotheremail@fakse.se
\${Name of placeholder\${1}}	Replaces the content <code>\${Name of placeholder\${1}}</code> with the first value matched by <i>Name of placeholder</i> . This is used when only the first value is required	<code>\${Price\${1}}</code>	12
		<code>\${Email\${1}}</code>	myemail@fakse.se
\${Name of placeholder\${n}}	Replaces the content <code>\${Name of placeholder\${n}}</code> with the n:th value matched by <i>Name of placeholder</i> . This is used when only the n:th value is required	<code>\${Price\${2}}</code>	<empty value>
		<code>\${Email\${2}}</code>	anotheremail@fakse.se
		<code>\${Email\${4}}</code>	<empty value>

Placeholder context

Placeholders can replace different data for different contexts (custom fields, automation, post functions).

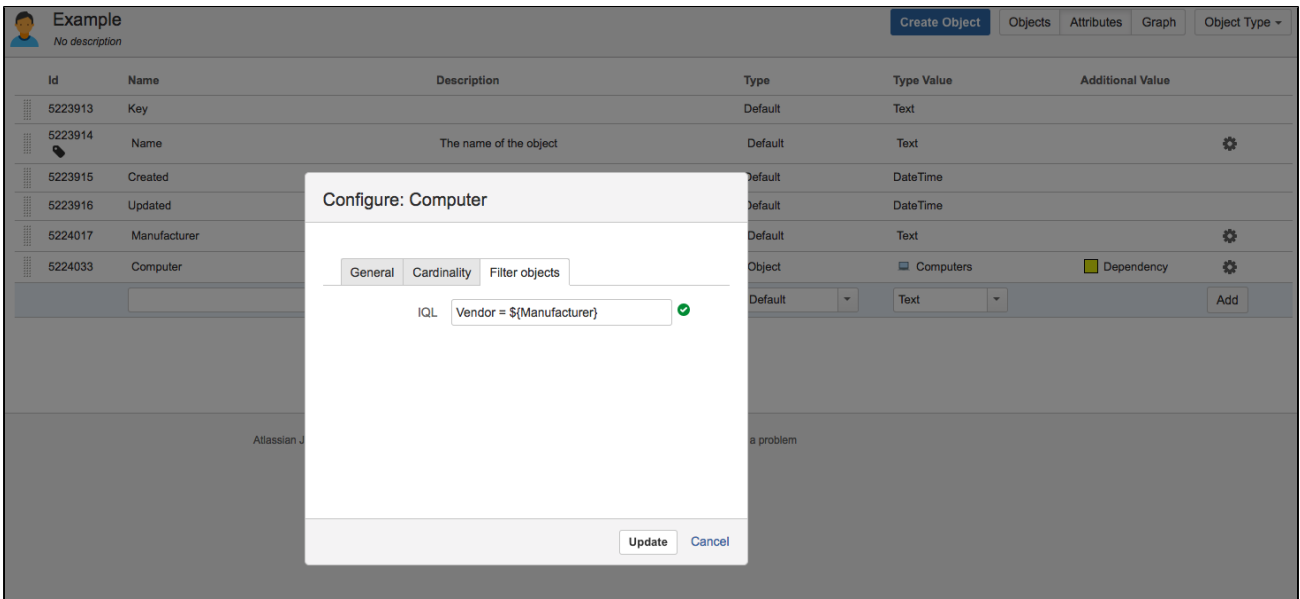
Automation rules

For more info on automation rules, see [Automation rules](#).

Placeholder name	Value	Description
Attribute Name	The attribute value	The value of the attribute with the same name as the placeholder. This will render the attribute in the human readable form.
Attribute Name.id	The attribute value as stored	The value of the attribute with the same name as the placeholder. This will render the attribute in the form that is stored. e.g. if the attribute is an object reference this will render the referenced object id. This will only work on attributes that are stored with another value than presented.
label	The label of the object	The value is replaced by the configured label for the object.
objectType	The object type name of the object	The value is replaced by the object type name.
objectTypeid	The object type id	The value is replaced by the id of the object type of the current object.
objectId	The objects id (not the key)	The integer that identifies this object.
Key	The key of the object	The key value of the current object.

Object create / edit

When creating or editing objects with object reference attributes you can specify an AQL that is used to limit the selection of objects. The placeholders available are all the attributes available in the create/edit dialog.



Placeholder name	Value	Description
Attribute Name	The attribute value	The value of the attribute with the same name as the placeholder. This will render the attribute in the human readable form
Attribute Name.id	The attribute value as stored	The value of the attribute with the same name as the placeholder. This will render the attribute in the form that is stored. e.g. if the attribute is an object reference this will be the id of the object that is referenced.

Here's a video if you'd like to see how this can be done:

Post-functions

For more info on post-functions, see [Post-functions](#).

The following post-functions support placeholders:

- Email user(s) based on attribute in a selected Assets object
- Set the value of an object attribute with a predefined value
- Assign object/s based on an AQL query

Custom fields

For more info on custom fields, see [Assets custom fields](#).

The standard Assets custom field "Assets Object/s" supports placeholders for any issue field data. This is configured in the "Issue Scope AQL" and can be constructed as the following example:

```
Organisation = ${customfield_12345.name} AND Location = ${Country} AND "Issue Type" like ${issuetype.name}
```

The placeholders will be replaced with the identifiers for the specific underlying objects. Remember to use IN in the AQL if you have multiple values.

Custom field key	Custom field name	\${placeholder.id} / \${placeholder.key}
------------------	-------------------	---

`\${customfield_XXXXX}`	<Custom Field Name>	The identifier	The label / name
--------------------------------	---------------------	----------------	------------------

The table below shows the keys for issue placeholders to use for Jira fields. Make sure to use IN for multi values to not break the AQL:

Issue key	Label / Description	Multiple	Type	`\${key.id}`	`\${key.label}`
key	The Issue Key	false	String	key	key
affectedVersions	Affected Versions	true	Version	Id	Name
assignee	Assignee	false	ApplicationUser	userKey	Display Name
description	Description	false	String	description	description
duedate	Due Date	false	Date	duedate	duedate
fixVersions	Fix Versions	true	Version	Id	Name
issuekey	Issuekey	false	String	key	key
environment	Environment	false	String	environment	environment
issuetype	Issue Type	false	IssueType	Id	Name
creator	Creator	false	ApplicationUser	userKey	Display Name
labels	Labels	true	String	labels	labels
components	Components	true	ProjectComponent	Id	Name
attachments	Attachments	true	Attachment	Id	filename
priority	Priority	false	Priority	Id	Name
pid	Project	false	Project	Id	Name
reporter	Reporter	false	ApplicationUser	userKey	Display Name
resolution	Resolution	false	Resolution	Id	Name
votes	Votes	true	Long	votes	votes
watchers	Watchers	true	Long	watchers	watchers
resolutiondate	Resolution Date	false	Date	resolutiondate	resolutiondate
created	Created	false	Date	created	created
updated	Updated	false	Date	updated	updated
status	Status	false	Status	Id	Name
summary	Summary	false	String	summary	summary

The *issuetype* placeholder won't work for Jira Service Management requests. You must use a custom field instead:

Wrong: issuetype

```
"Related Jira Projects" = currentProject() AND "Related Jira Issue Types"."Name" in (${issuetype.label})  
AND Status = "Active"
```

Right: customfield

```
"Related Jira Projects" = currentProject() AND ("Related Jira Issue Types"."Name" in (${issuetype.  
label}) OR "Related Jira Issue Types"."Name" in (${customfield_10654})) AND Status = "Active"
```


Assets AQL functions

We provide a number of JQL functions that can be in handy when searching for issue in Jira. Below we will describe how they work and what you can do with them. If you have any request for additional JQL functions, please let us know and we will add it to our backlog.

What is issueFunction below?

In the JQL examples below you will find something called **customField**. It is your custom field that you may add to your Jira instance. Remember that the name is of your own choice, so use the ones that you have created.

aqlFunction

 using the AQL function with object schema or specifying an object type in the AQL will increase performance by a lot. If the AQL function is used when calculating JSD SLA's for instance, try to limit the AQL by adding object type name, schema id or object key since this will decrease the execution time of the query.

aqlFunction(aql)

aql is the [AQL](#) to filter on.

Example:


```
customField in aqlFunction("Customer.\"Account Manager\" = \"Mathias Edblom\"")
```

or

```
customField in aqlFunction("Host.Virtual = true")
```

This will filter any issues where an application is reported and where the application is installed on a virtual server.

aqlFunction(object schema name/id, aql) (Deprecated)

 This one is deprecated since the object schema/s are determine by the custom field configuration it self. This possibility will be removed in future releases.

Object Schema Name/ID is the name (or id) of an object schema.

aql is the [AQL](#) to filter on.

attributeValue (Deprecated)

 This JQL function is deprecated. Use the **aqlFunction** instead!

attributeValue(attribute, operator, value)

Attribute is the name of an object type attribute.

Operator can be one of the following: "=", "!=", "<", "<=", ">", ">=", "like", "startswith", "endswith"

Value is the value to search on.

Example:


```
customField in attributeValue("salary", ">", "70000")
```

This will return all issues with an insight object connected and the object has an attribute "salary" and the value is more than 70000.

Example:

```
customField in attributeValue("Expiration Date", "<", "28/Jun/14")
```

This will return all issues with an insight object connected and the object has an attribute "Expiration Date" and the date is earlier than 28/Jun/14.

 You may use **now()** to use current date. Use **now(90)** to add 90 days from today and similar query's like **now(-30)**.

Example:


```
customField in attributeValue("Name", "like", "joh")
```

This will return all issues with an insight object connected and the object has "joh" somewhere in the name.

Example:

```
customField in attributeValue("Owner", "=", "Mathias Edblom")
```

This will return all issues with an insight object connected and the Jira user "Mathias Edblom" is the owner of the object/s.

 For User type, you may use "**currentUser()**" to filter on the current user. Ex: customField in **attributeValue("Owner", "=", "currentUser()")**

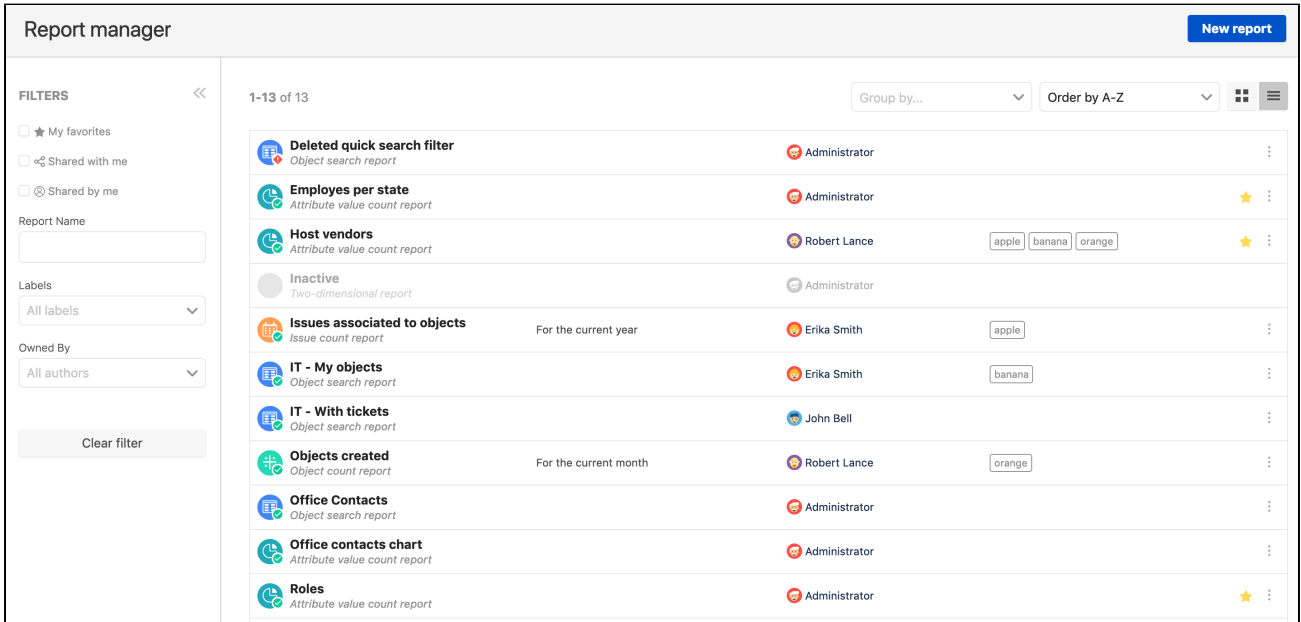
 Note that attribute name are **case-sensitive** in the attributeValue function!

Working with reports

You can use dynamic reports to view information about your object schema in a variety of formats. You can create a report based on any of the default report types.

Creating a report

1. In the top navigation bar, select **Assets > Reports**.
2. Select **New Report**.



Once you start creating reports, they'll appear in the report manager. Here you can order and filter them, or do many other things, such as printing, sharing, and syncing your reports. To see the list of available actions for your reports, select **Options** next to each report.

Report types

Here are all report types that you can create in Assets:

Report	Description	Example																		
Attribute value count report	Pie chart showing how the objects of an object type are distributed based on attributes.	<p>The screenshot shows the configuration for a 'New Attribute value count report'. The 'Name' field is 'Components by Vendor'. The 'Object schema' is 'AM', 'Object type' is 'Component', and 'Attribute' is 'Vendor'. The 'Object filter (SQL)' is 'Name LIKE John Smith'. The preview shows a donut chart with the following data:</p> <table border="1"> <thead> <tr> <th>Vendor</th> <th>Percentage</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>INTEL</td> <td>55%</td> <td>11</td> </tr> <tr> <td>Western Digital</td> <td>5%</td> <td>1</td> </tr> <tr> <td>Kingston</td> <td>15%</td> <td>3</td> </tr> <tr> <td>Seagate</td> <td>20%</td> <td>4</td> </tr> <tr> <td>Corsair</td> <td>5%</td> <td>1</td> </tr> </tbody> </table>	Vendor	Percentage	Count	INTEL	55%	11	Western Digital	5%	1	Kingston	15%	3	Seagate	20%	4	Corsair	5%	1
Vendor	Percentage	Count																		
INTEL	55%	11																		
Western Digital	5%	1																		
Kingston	15%	3																		
Seagate	20%	4																		
Corsair	5%	1																		

<p>Attribute value report</p>	<p>Chart showing attributes totals reported for one or more objects of a specific type.</p>	
<p>Catalog report</p>	<p>Catalog of objects and their attributes.</p>	
<p>Issue count report</p>	<p>Line or bar chart showing historical totals of issues reported for one or more object types or objects.</p>	
<p>Map report</p>		

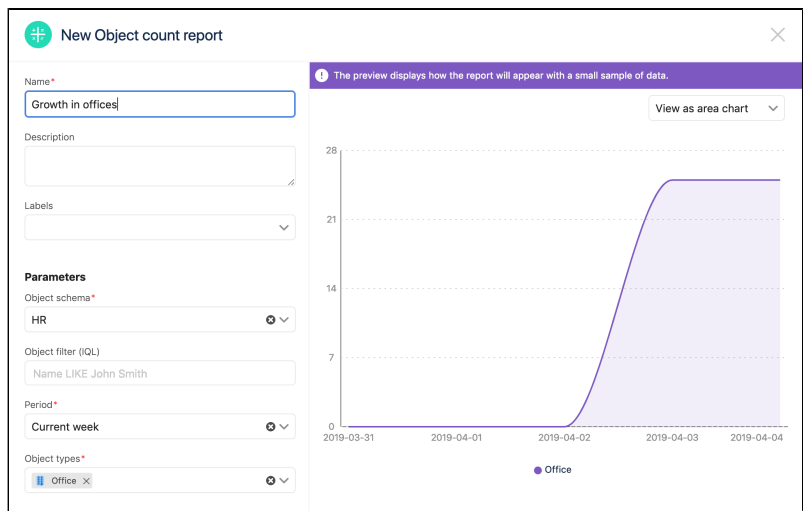
World map showing the location of objects based on an attribute with geographical data. The objects must have a text attribute with a comma-separated latitude and longitude, for example `-33.8,151.2` (Sydney).

The screenshot displays the 'Location of servers' map report interface. At the top left is a purple icon of a document with a list. The main title is 'Location of servers' in a large, bold, dark blue font, with the subtitle 'Map report' in a smaller, grey font below it. To the right of the title are three icons: a share icon, a star icon, and a vertical ellipsis menu icon. Below these icons, the 'Owner' is listed as 'Friendly Robot' with a circular profile picture. The 'Synchronize' frequency is set to 'Once', and the 'Last update' is 'at 11:15 am'. Under the 'Parameters' section, 'Object schema' is 'ITSM' and 'Object type' is 'Servers'. The 'Location' attribute is also listed as 'Location'. On the right side of the interface is a vertical map area with a light blue background and a small map showing landmasses. A vertical toolbar on the right edge of the map area contains three icons: a plus sign for zoom in, a minus sign for zoom out, and a square with an 'X' for full screen. At the bottom right of the map area, the text 'Island area' is visible.



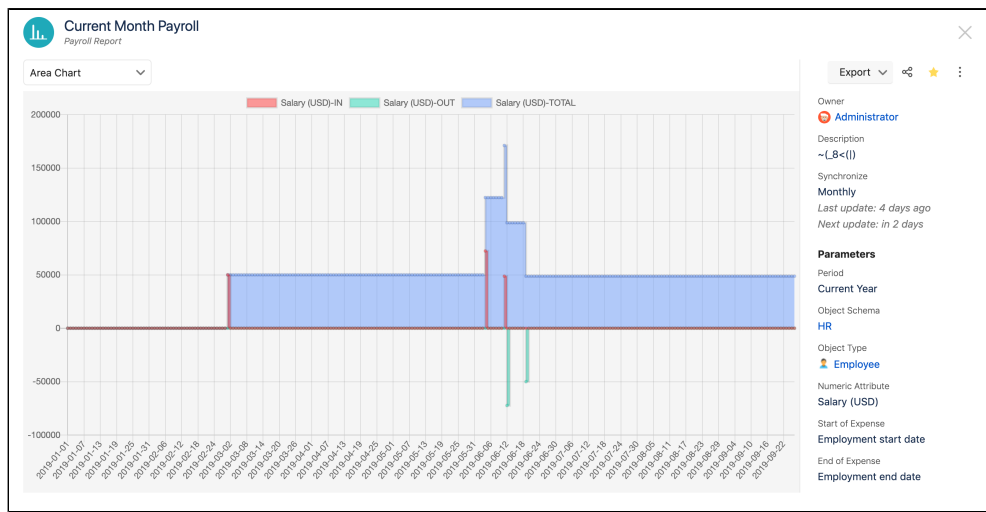
Object count report

Line chart showing historical totals of objects for one or more object types.



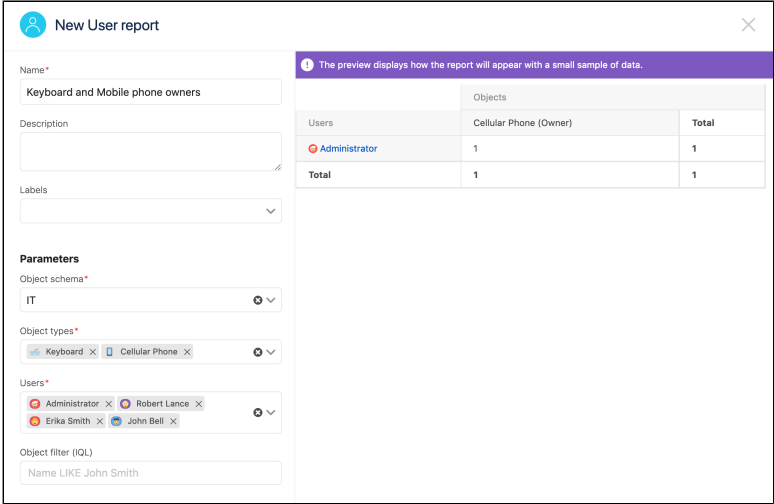
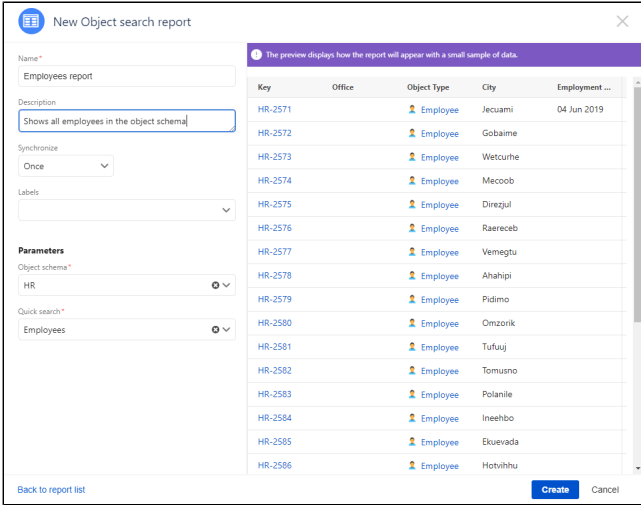
Payroll report

Chart showing payroll expenditures.



Two-dimensional report

Returns a table showing the objects of an object type and their attributes.

<p>User report</p>	<p>Returns a table showing the total of one or more objects assigned to one or more users.</p>	 <p>The screenshot shows the 'New User report' configuration window. The 'Name' field is 'Keyboard and Mobile phone owners'. The 'Description' field is empty. The 'Labels' field is empty. The 'Parameters' section includes: 'Object schema' set to 'IT', 'Object types' including 'Keyboard' and 'Cellular Phone', 'Users' including 'Administrator', 'Robert Lance', 'Erika Smith', and 'John Bell', and 'Object filter (JQL)' set to 'Name LIKE John Smith'. The preview table shows:</p> <table border="1" data-bbox="778 232 1246 331"> <thead> <tr> <th>Users</th> <th>Objects</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>Administrator</td> <td>Cellular Phone (Owner) 1</td> <td>1</td> </tr> <tr> <td>Total</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	Users	Objects	Total	Administrator	Cellular Phone (Owner) 1	1	Total	1	1																																																																												
Users	Objects	Total																																																																																					
Administrator	Cellular Phone (Owner) 1	1																																																																																					
Total	1	1																																																																																					
<p>Object search</p>	<p>Report is based on a saved quick search (filter) from the Assets Object Search View. It's a table showing the objects of an object type and their attributes. When you open an object search report, you can select the Quick search link to view the filter that generates the table.</p>	 <p>The screenshot shows the 'New Object search report' configuration window. The 'Name' field is 'Employees report'. The 'Description' field contains 'Shows all employees in the object schema'. The 'Synchronize' dropdown is set to 'Once'. The 'Parameters' section includes: 'Object schema' set to 'HR' and 'Quick search' set to 'Employees'. The preview table shows:</p> <table border="1" data-bbox="730 757 1114 1160"> <thead> <tr> <th>Key</th> <th>Office</th> <th>Object Type</th> <th>City</th> <th>Employment ...</th> </tr> </thead> <tbody> <tr><td>HR-2571</td><td></td><td>Employee</td><td>Jecumi</td><td>04 Jun 2019</td></tr> <tr><td>HR-2572</td><td></td><td>Employee</td><td>Gobaime</td><td></td></tr> <tr><td>HR-2573</td><td></td><td>Employee</td><td>Wetcurhe</td><td></td></tr> <tr><td>HR-2574</td><td></td><td>Employee</td><td>Mecocob</td><td></td></tr> <tr><td>HR-2575</td><td></td><td>Employee</td><td>Direzul</td><td></td></tr> <tr><td>HR-2576</td><td></td><td>Employee</td><td>Raereceb</td><td></td></tr> <tr><td>HR-2577</td><td></td><td>Employee</td><td>Vernegtu</td><td></td></tr> <tr><td>HR-2578</td><td></td><td>Employee</td><td>Ahahipi</td><td></td></tr> <tr><td>HR-2579</td><td></td><td>Employee</td><td>Pidimo</td><td></td></tr> <tr><td>HR-2580</td><td></td><td>Employee</td><td>Omzonik</td><td></td></tr> <tr><td>HR-2581</td><td></td><td>Employee</td><td>Tufuuj</td><td></td></tr> <tr><td>HR-2582</td><td></td><td>Employee</td><td>Tomusno</td><td></td></tr> <tr><td>HR-2583</td><td></td><td>Employee</td><td>Polanile</td><td></td></tr> <tr><td>HR-2584</td><td></td><td>Employee</td><td>Ineehbo</td><td></td></tr> <tr><td>HR-2585</td><td></td><td>Employee</td><td>Ekuevada</td><td></td></tr> <tr><td>HR-2586</td><td></td><td>Employee</td><td>Hotvihu</td><td></td></tr> </tbody> </table>	Key	Office	Object Type	City	Employment ...	HR-2571		Employee	Jecumi	04 Jun 2019	HR-2572		Employee	Gobaime		HR-2573		Employee	Wetcurhe		HR-2574		Employee	Mecocob		HR-2575		Employee	Direzul		HR-2576		Employee	Raereceb		HR-2577		Employee	Vernegtu		HR-2578		Employee	Ahahipi		HR-2579		Employee	Pidimo		HR-2580		Employee	Omzonik		HR-2581		Employee	Tufuuj		HR-2582		Employee	Tomusno		HR-2583		Employee	Polanile		HR-2584		Employee	Ineehbo		HR-2585		Employee	Ekuevada		HR-2586		Employee	Hotvihu	
Key	Office	Object Type	City	Employment ...																																																																																			
HR-2571		Employee	Jecumi	04 Jun 2019																																																																																			
HR-2572		Employee	Gobaime																																																																																				
HR-2573		Employee	Wetcurhe																																																																																				
HR-2574		Employee	Mecocob																																																																																				
HR-2575		Employee	Direzul																																																																																				
HR-2576		Employee	Raereceb																																																																																				
HR-2577		Employee	Vernegtu																																																																																				
HR-2578		Employee	Ahahipi																																																																																				
HR-2579		Employee	Pidimo																																																																																				
HR-2580		Employee	Omzonik																																																																																				
HR-2581		Employee	Tufuuj																																																																																				
HR-2582		Employee	Tomusno																																																																																				
HR-2583		Employee	Polanile																																																																																				
HR-2584		Employee	Ineehbo																																																																																				
HR-2585		Employee	Ekuevada																																																																																				
HR-2586		Employee	Hotvihu																																																																																				

Providing help in multiple languages

You can meet your customers language preferences by adding additional languages to your service project. This translates the language they receive when theyre using your portal and help center to their preferred language.

 You need to be a Jira Service Management project admin to change the language settings.

What is language support?

When you create a serviceproject, itdisplays in the default language inherited from your Jira instance. To meet your customer's language preferences, you can add multiple languages to their service projects.

You can translate your:

- request types (names, descriptions, fields, and groups)
- portal names
- announcements (login, help center, and portal)
- customer notifications

Limitations

Here are some things that aren't included right now:

- Linking a Confluence space to a chosen language
- Translating the customer satisfaction survey
- Translating the Approve and Reject buttons for approvals
- Searching for translated request types and portals on the customer portal

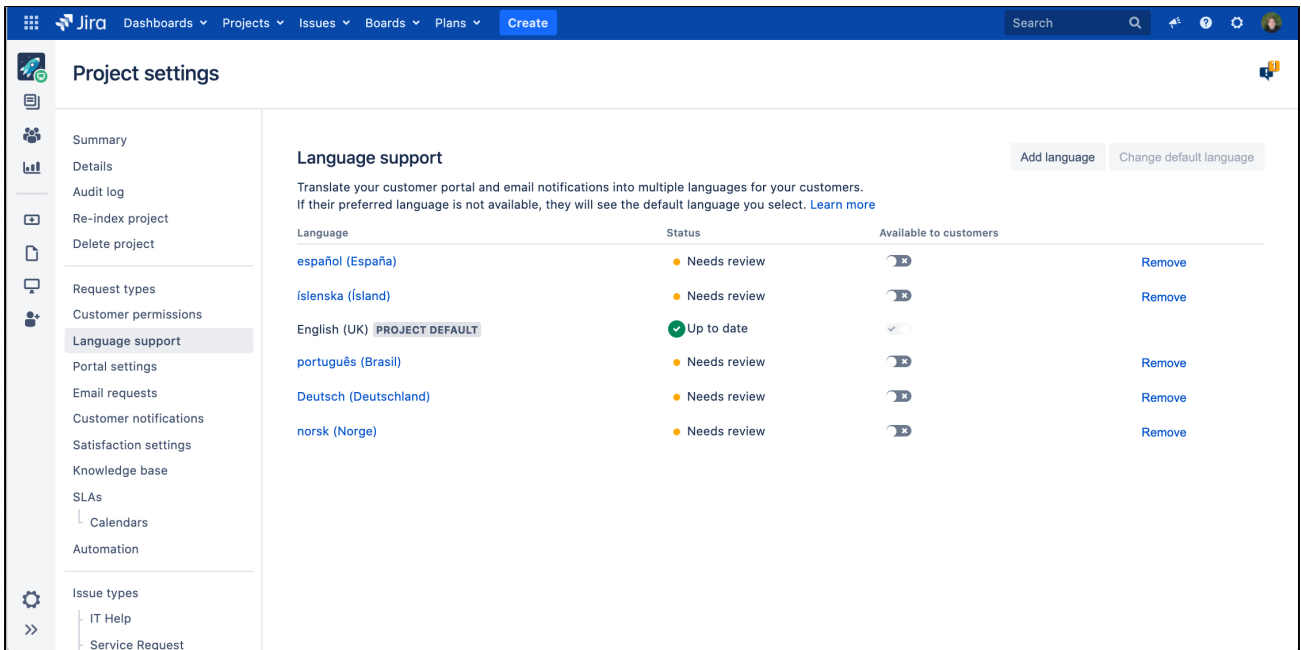
Add languages to your service project

Your team can add and edit translations based on the languages of your service project customers. Adding translations is especially important for customized content, such as help center announcements, which will otherwise appear in your default service project language.

To add languages and translations:

1. From your service project, go to **Project settings > Language support**.
2. Select **Add language**.
3. Select the language and select **Add**.

Your languages appear in the languages table. New languages dont come with ready translations, so youll need to edit and update them. We disable newly added languages by default. We do this so you can quality check your new language and its translations before customers see them.



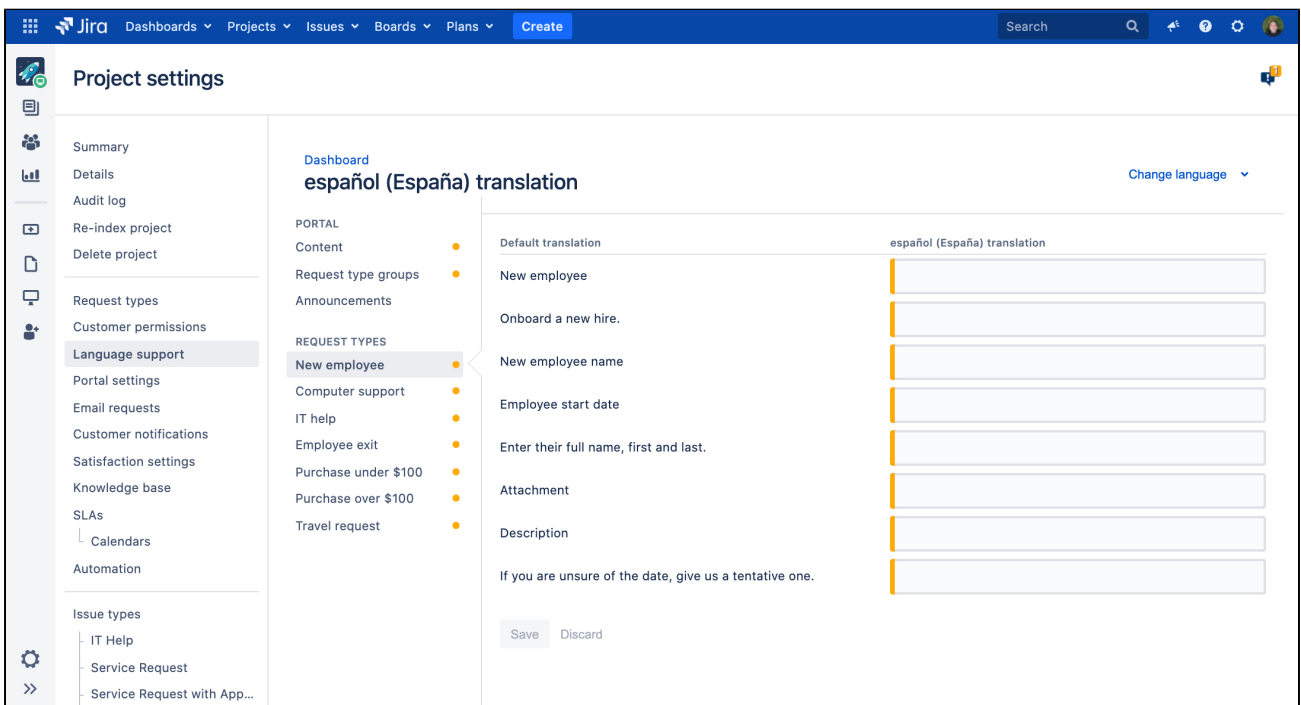
Review or update translations

You can add custom translations to your service project to enhance your customers experience. Custom content may include translations of request types, portal names, and announcements.

To review or update custom language translations:

1. From your service project, go to **Project settings > Language support**.
2. Select the language's entry in the **Language** column.
3. Review or update the languages content.
4. Select **Save**.

We group your customer-facing content into sections in the translation space. Use the service project sidebar to jump between sections. For most sections, the service project default language appears in the left column and the translated text appears in the right.



Announcement banner translations

To prevent inconsistencies when you delete your announcement banner for the help center or the customer portal, we'll check if there are any translations attached to it. If there are, the translations for the deleted fields will be cleared and you'll see a warning message to confirm the change.

When you update the announcement banner, we'll keep the existing translations. However, if there are any translations attached to it, you'll see a warning message for reviewing existing translations.

To review your announcement translations:

- For global announcement translations, go to **Project settings > Language support**. We recommend consistently using only one selected project to manage all your global announcement translations.
- For customer portal announcement translations, select the project that is linked to the specific portal. Next, go to **Project settings > Language support**.

The screenshot shows the 'Project settings' page in Jira Service Management. The left sidebar contains a navigation menu with 'Language support' selected. The main content area is titled 'Deutsch (Deutschland) translation' and displays a table of announcement translations. The table has two columns: 'Default translation' and 'Deutsch (Deutschland) translation'. The rows include: 'This is a login subject' (translated: 'Dies ist ein Login-Betreff'), 'This is a login message' (translated: 'Dies ist eine Anmeldenachricht'), 'This is a help center announcement subject' (translated: 'Dies ist ein Thema für Ankündigungen in der Hilfe'), 'This is a help center announcement message' (translated: 'Dies ist eine Ankündigungsnachricht des Hilfezentrums'), 'This is a portal announcement subject' (translated: 'Dies ist ein Portalankündigungsthema'), and 'This is a portal announcement message' (translated: 'Dies ist eine Portalankündigungsnachricht'). At the bottom of the table are 'Save' and 'Discard' buttons.

Change your service project's default language

We recommend setting the service projects default language to match the language of the majority of your customers. Setting your service projects default language to be the same as your customers, allows them to effectively use their service project.

Customers will see the service projects default language if the project admin hasn't added their preferred language or the preferred language is disabled.

To set the default service project language:

1. From your service project, go to **Project settings > Language support**.
2. Select your preferred language.
3. Review your language's translations and confirm they are up to date.
4. Select **Save**.
5. Toggle the slider so the language is **Available to customers**.
6. Select **Change default language**.
7. Select your required language.
8. Select **Change**.

Stop supporting a language

If you'd like to stop supporting a language, we recommend you disable rather than remove it. That way, you can keep up with translations in case you re-enable it. Removing a language would also remove its translations.

To stop supporting a language:

1. From your service project, go to **Project settings > Language support**.
2. Find the language's entry in the **Language** column.
3. In the **Available to customers** column, toggle the slider off.

As a result of disabling a language, customers whose preferred language is no longer supported will see the portal text in their preferred language (for example, **Log in** or **Profile**), and the serviceproject request types and notifications in the service project default language.

Let customers know how to set their language preference

Customers can change their own language preference in their help center profile.

To assist customers to change their language preference, recommend the following:

1. From the help center, select the avatar.
2. Select **Edit preference** under **Preferences**.
3. In the **Language** dropdown, select the preferred language.
4. Select **Save**.

Translate customer notifications

You can also add languages for notifications sent to customers. In this case, you wont be able to provide custom translationsthe notifications will use the translations provided with Jira.

- See [Translate Jira Service Management notifications](#).

Collecting customer satisfaction (CSAT) feedback

Measuring customer satisfaction can help you better understand your customers and improve service levels.

Jira Service Management provides a simple, built-in mechanism to collect customer feedback. Key features include:

- Simple customer workflow to provide feedback on resolved issues
- Customer satisfaction scores are visible within resolved issues and on agent queues for resolved issues
- Single-click to view customer satisfaction report for a service project
- Easily create and view custom reports and trend graphs based on satisfaction scores.

Customer feedback data can be used to identify strengths and weaknesses in the service quality, engage and motivate the team to improve satisfaction scores, and provide mentoring and training where required.

On this page:

- [Enabling the customer satisfaction feature](#)
- [Viewing and reporting on customer feedback](#)

Enabling the customer satisfaction feature

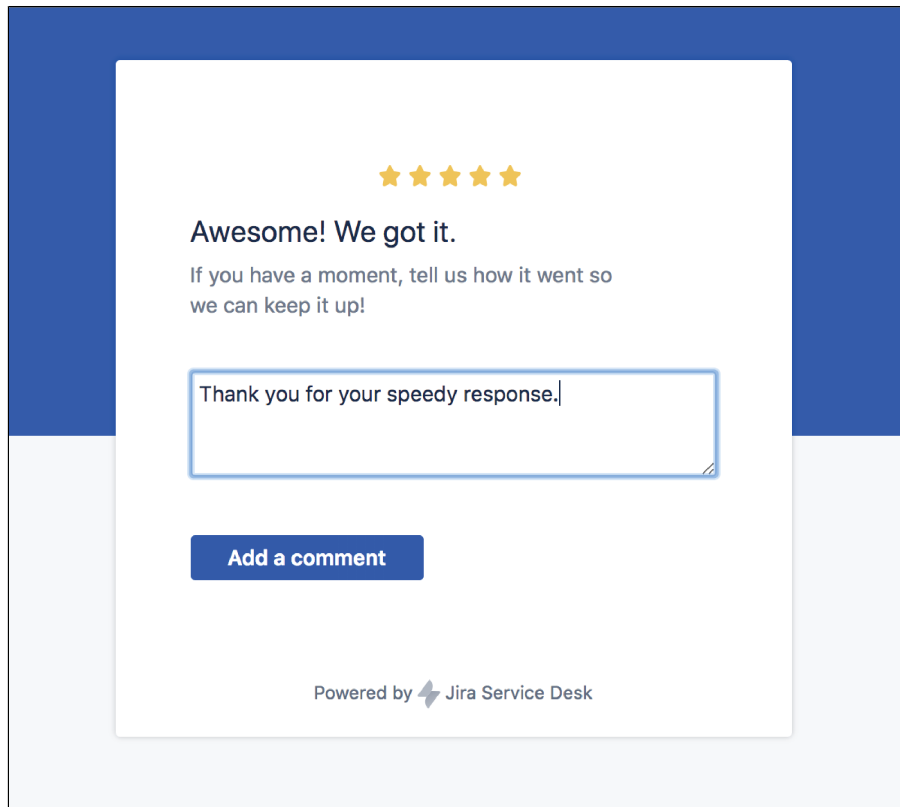
This feature is enabled by default for new service projects; however, it must be enabled for each existing service project. To enable customer satisfaction settings for an existing project:

1. Log in as a service project administrator.
2. Select the service project you wish to configure.
3. Select **Project settings** > **Satisfaction settings**.
4. Enable **Collect customer satisfaction feedback**.
5. Optionally, edit the **Question** phrase to suit your environment. This phrase appears in the resolved issue notification message that customers see.

You can send satisfaction rating email notifications to your customers whenever a request is resolved. To do so, you need to enable the Request resolved rule:

1. Go to **Project settings** > **Customer notifications**.
2. Next to Request resolved select **Edit**.
3. Select **Enable**, then select **Save**.

When you enable satisfaction settings for a service project and enable the Request resolved rule, the resolved issue notifications will contain a satisfaction rating scale. Customers can select the rating scale to indicate their level of satisfaction. A confirmation page is displayed on the customer portal, where they can change the rating, and optionally provide any additional comments that they would like to convey to the team.



Viewing and reporting on customer feedback

Customer satisfaction scores and comments are displayed in the issue view for resolved issues. Agents can also view the satisfaction scores on their own recently resolved queues.

Service project administrators and agents can view the default Satisfaction report, which displays the average customer satisfaction scores for the team.

Customer satisfaction				
<p>2.6 Average rating</p>				
Rating	Comment	Key	Agent	Received
★★★★★	Thanks for coming down to my desk to help me print in A3	IT-1384	Will Turner	Yesterday
★★★★★		IT-997	José Vela	2 days ago
★☆☆☆☆	I had to wait 3 days for a solution. I replied on Wednesday but nobody got back to me until this morning.	IT-1216	Mia Kennedy	3 days ago

1-3 of 3

Past 7 days

A service project administrator can also create and view [custom reports](#) analyzing satisfaction trends. Agents can also view any custom satisfaction reports created for their service projects.



Useful examples of custom reports include:

- A trend graph of the average satisfaction rating for a specific period to view changes in service levels.
- Satisfaction scores based on the type of service request. This would identify issues for which the team could provide knowledge articles.
- Satisfaction scores for an individual agent compared to team scores to help identify agents who would benefit from further training.

Jira Service Management best practices

Check out the following best practice articles:



- [Best practices for designing the customer portal](#)
- [Best practices for IT teams using Jira Service Management](#)
- [Best practices for software teams using Jira Service Management](#)

Best practices for designing the customer portal

Every service project comes with an online portal that your customers can use to interact with your service team.

Here are some best practices on how to design a customer portal that makes it easy to find help, and work effectively.

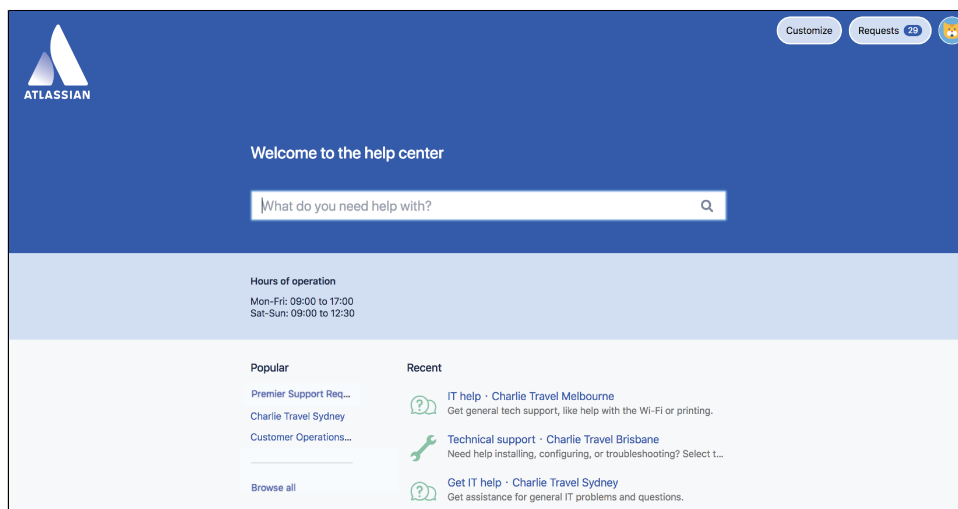
On this page:

- [Brand your help center](#)
- [Brand your customer portal](#)
- [Help customers find request types](#)
- [Group similar request types together](#)
- [Set up a knowledge base](#)

Brand your help center

Customers can use the global help center to browse all the service projects they have access to. You can brand your help center in a number of ways:

- Add a company logo
- Name your help center
- Add a banner image across the top
- Change the color of some components.

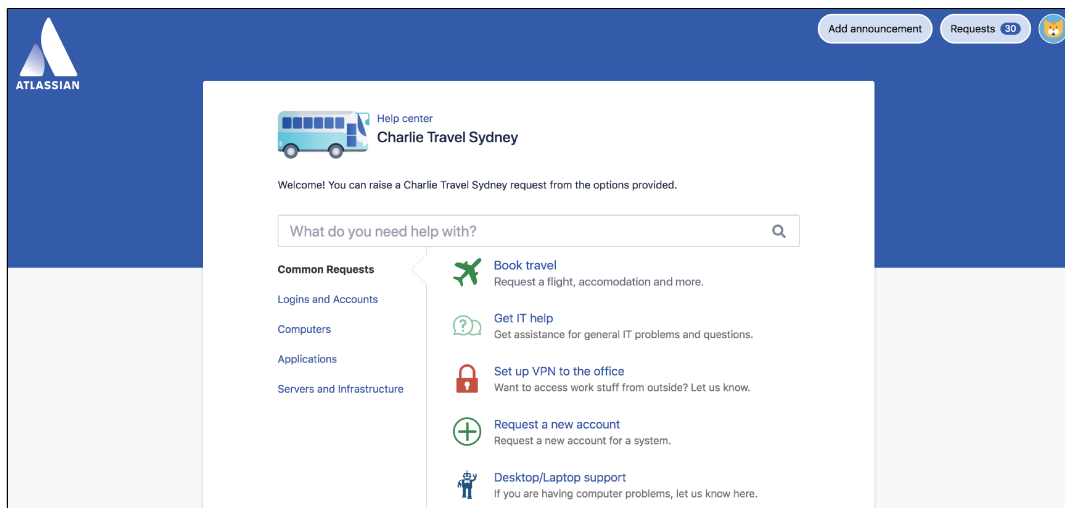


What's more, your customer portals will inherit these design changes making the experience consistent across all of your service project channels.

To learn how to do this check out [Using the help center](#).

Brand your customer portal

Add a name and logo to your customer portal so that people know they're sending requests to your team and organization. It will look something like this:



To learn how to do this check out [Configuring the customer portal](#).

Help customers find request types

You can set up request types in a way that makes it easier for your customers to find them. Here's how:

- Name your request type something that your customers will understand. For example, *Access to a system* instead of *VPN access*.
- Use different icons for each request type so customers can recognise *IT help* from, say, *Book a flight*.
- Add contextual help. For example, say what size and format is accepted in the attachment field.
- Use examples in your request type descriptions. For example, *If you need a software license such as Microsoft Office, raise a request here*.
- Link to helpful information in the request type description. For example, if you have a list of available Microsoft Office license numbers on your intranet, add a link so they don't need to open a request.

To learn how to do this check out [Setting up request types](#).

Group similar request types together

If you have a lot of request types (more than seven), we recommend grouping some of them together to help customers find what they need. Grouped request types appear as tabs in your customer portal.

Project settings

Summary

Details

Re-index project

Delete project

Request types

Customer permissions

Portal settings

Email requests

Customer notifications

Satisfaction settings

Knowledge base

SLAs

Request

GROUPS

⋮ Comm

⋮ Logins

⋮ Comp

⋮ Applic

⋮ Server

⋮ Tester

Hidde

+ Add g

To add groups:

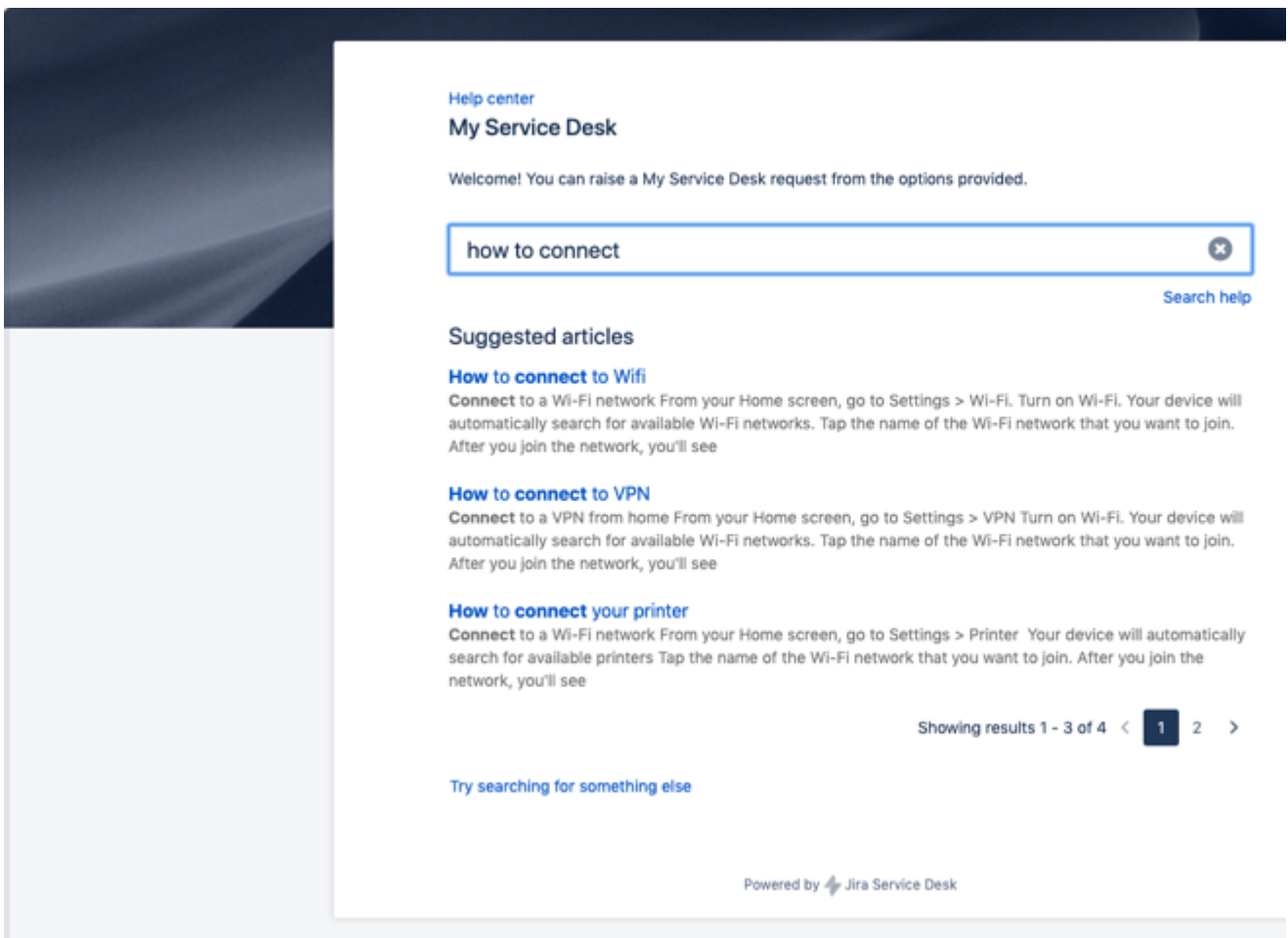
1. Select **+Add group** from the sidebar.
2. While viewing a group, select **Add existing request type** to add your request types to it.
3. Change the order that the groups appear on the customer portal, by drag and drop.

To learn more about request types, check out [Setting up request types](#).

i To prevent a request type from showing on the customer portal, move it to the **Hidden from portal** group, underneath the other groups. Removing a request type won't impact requests that have already been created.

Set up a knowledge base

Over time, your service team will accumulate a lot of knowledge that can be shared with your customers so that they can solve their own problems without needing to open a request. You can set up a knowledge base by integrating Jira Service Management with Confluence.



To learn how to do this check out [Set up a knowledge base for self-service](#).

Best practices for IT teams using Jira Service Management

Information Technology Infrastructure Library (ITIL) is a framework for [ITSM](#) (IT service management). ITIL recommendations have been an industry standard for 20 years. Adopting these practices takes time. You can seek formal training on how to make ITIL recommendations work best for your business.

If you're interested in more general information about ITSM or other guides, [check out our ITSM resources](#).

We recommend starting with processes that are essential to your business goals. Jira Service Management provides workflows for IT teams in our IT Service Management project template. We created the following ITIL workflow add-ons, available in the Atlassian Marketplace:

- [Change management for Jira Service Management](#)
- [Jira Service Management incident management](#)
- [Problem management for Jira Service Management](#)
- [Service requests for Jira Service Management](#)

Jira Service Management is configurable. We recommend you start with an ITIL framework and then adjust to your specific business needs.

What follows is an overview of some best practices for your IT service desk. This guide covers:

- [Fulfilling service requests with your IT service desk](#)
- [Managing changes with your IT service project](#)
- [Managing incidents with your IT service desk](#)
- [Managing problems with your IT service desk](#)
- [Calculating priority automatically](#)
- [Manage changes with your IT service management project 1](#)

Use the IT Service Management template

Only Jira administrators can create projects.

To create a project using the IT Service Desk template:

1. Select **Projects > Create Project**.
2. Choose the **IT Service Management** template and select **Next**.
3. Name your project.
4. Select **Submit**.

Provide a robust service catalog

Using simple forms, your customers can do a lot of the early leg work for your service project. Jira Service Management provides them with clear and concise options for requesting help. A robust service catalog makes sure that service requests are prioritized and get to the correct service agent. And, your agents get the information needed to fulfill the request before beginning work.

The IT Service Management template comes with a service catalog of common requests. We call these forms "request types" and you can customize them to suit your needs.

To start, identify the most common and urgent IT tasks for your service project. Double check that there's a corresponding request type for each of these.

Go to **Project settings > Request types** to view or edit your service catalog. The entries there appear in your customer portal. [Read more about request types](#).

Help your customers serve themselves with a knowledge base

Mature IT service desks solve common problems without ever seeing a ticket. We recommend providing your customers a knowledge base.

Link your Jira Service Management site to a Confluence knowledge base. Keep a record of known solutions and two important things happen:

1. Your customers find and view relevant articles when they search the customer portal. They may find the answers they need without ever raising a request.
2. Your service agents can find relevant articles when working on issues. This saves them time hunting down answers or workarounds to common IT requests.

Learn more at [Set up a knowledge base for self-service](#).

Start with ITIL recommended workflows and adapt to your needs

ITIL recommendations are a framework, a set of ITSM best practices meant for you to adapt from and grow.

In Jira Service Management, we associate your service catalog with workflows by assigning a request type to an underlying *issue type*. Our recommendations for IT teams use four ITIL-inspired workflows. To learn more about how these issue types and workflows work to streamline your service project, check out our guides for handling:

- [Fulfilling service requests with your IT service desk](#)
- [Managing changes with your IT service project](#)
- [Managing incidents with your IT service desk](#)
- [Managing problems with your IT service desk](#)
- [Calculating priority automatically](#)
- [Manage changes with your IT service management project 1](#)

These allow you to have many customer request forms that follow the same workflow. For example, new hardware requests and password resets use the same service request workflow.

Our default workflows make your service project effective, out of the box. You can customize them as you go, scaling to the needs of your business.

Fulfilling service requests with your IT service desk

The scope of service requests in Information Technology Infrastructure Library (ITIL) is large. Tasks can range from resetting a password to onboarding a new hire. Service requests include customer comments, complaints, or other requests for information.

The IT Service Desk template comes with a few pre-built service request types. We set these up to help your service desk agents handle common service requests.

The service request fulfillment process:

- manages customer expectations
- speeds up request resolution
- standardizes any approval processes

Effective service request management reduces the bureaucracy and cost of maintaining IT services.

This page describes some best practices for fulfilling service requests using Jira Service Management. You may seek formal training in how to make ITIL recommendation work best for your business.

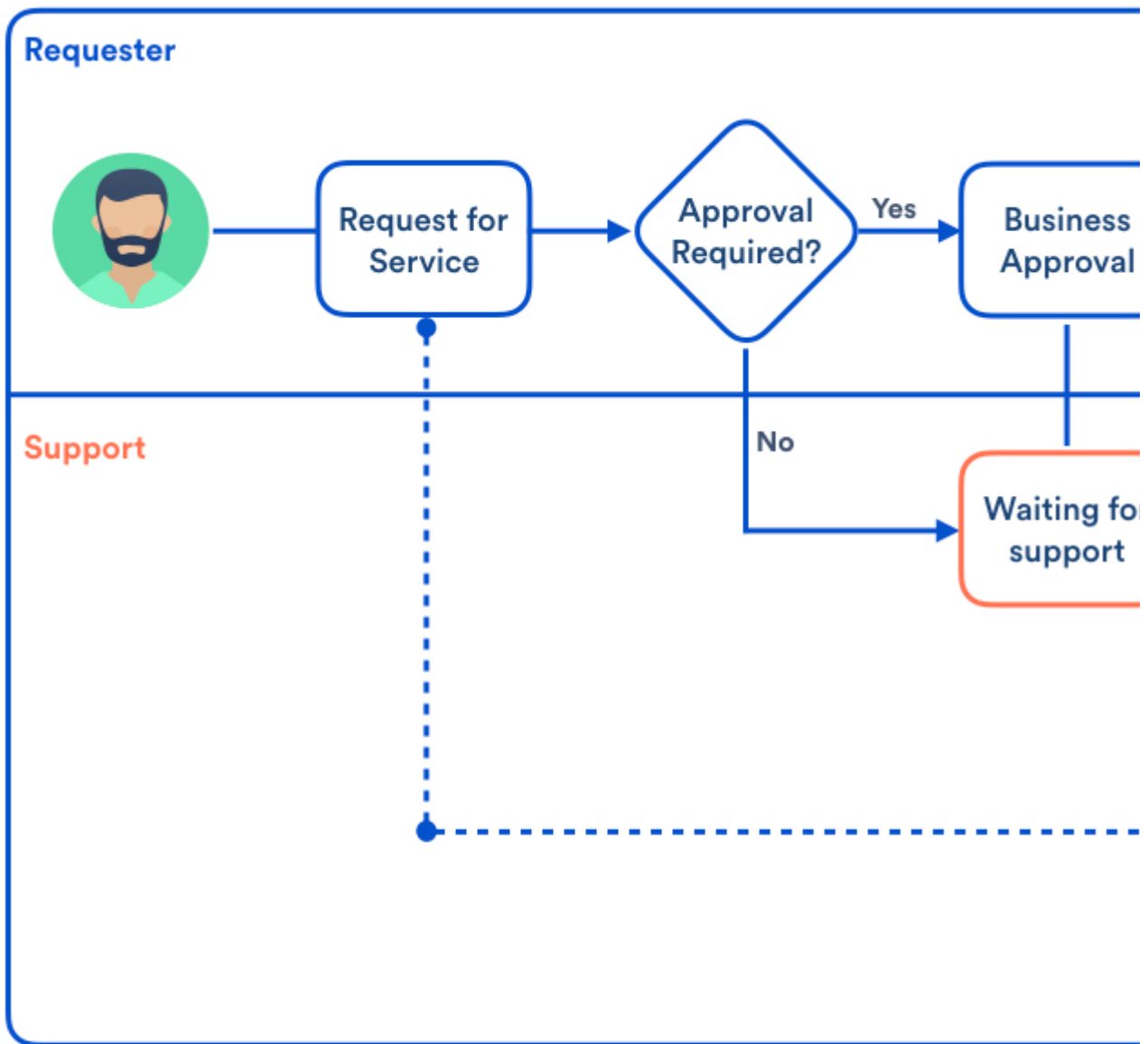
Service request fulfillment process

The information needed to capture and resolve service requests varies. But, you can standardize the process for fulfilling these requests.

The IT Service Desk template associates certain requests with a service request fulfillment workflow. We set up the workflow to complement the following service request fulfillment process. Use it as a jumping off point for your service desk.

The service request fulfillment process, in brief:

Service Request Fulfillment



Service Request Values



1. A customer requests help from your service catalog or via email.
2. The service desk assesses the request alongside pre-defined approval and qualification processes. If needed, they send the request for financial or business approval.
3. A service desk agent works to fulfill the service request, or forwards the request to someone who can.
4. After resolving the request, the service desk closes the ticket. The agent consults the customer to make sure they are satisfied.

Setup for service request fulfillment in Jira Service Desk

Configure the workflow and fields with the service request workflow add-on

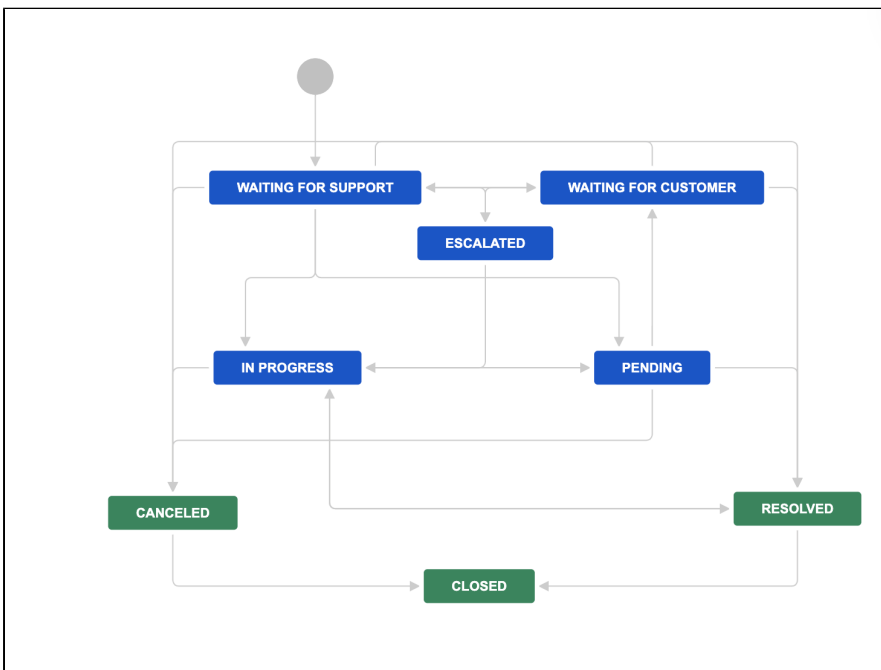
We used the ITIL framework for change management to build a workflow for Jira Service Management: <https://marketplace.atlassian.com/plugins/com.atlassian.servicedesk.servicerequest/server/overview>.

You can use this add-on as a template to help you build your own service request fulfillment process.

To use the workflow from the Marketplace:

1. Log in as a user that has the Jira administrator global permission, and follow the instructions listed here to [import a workflow](#).
2. To add the workflow fields to your change issues, activate the screen by following the instructions here: [Defining a screen](#).

Service request fulfillment workflow



Tips for creating service request forms on your portal

- Begin with the most common requests. Choose ones that are simple and easy to fulfill. This delivers immediate value to customers. It allows the service desk team to learn as they build out future phases of the service catalog.
- Document all the requirements for a service request before adding it to the catalog. These include question data, the approval process, fulfillment procedures, the fulfillment team, process owners, SLAs, reports, and so on. This allows the IT team to better manage the request type over time.
- Capture the data needed to start fulfilling the request. But, don't overload the customer with too many questions.
- Work with stakeholders to standardize the approval process, where possible. For example, pre-approve all requests for new monitors. Or, assign software approvals to the customer's manager.
- Document any knowledge base information that might allow customers to service their own request. Record this in a linked Confluence space. If you do, customers can view articles while they search your portal. [Read more about creating a knowledge base](#).
- Review your team's performance in fulfilling requests. Adjust your SLAs, requirements, and training to improve customer satisfaction.
- Create reports to help manage the lifecycle of a service request offering. These trends can uncover forms that are no longer needed, too complex, or insufficient.

[Read more about creating reports](#).

Default form fields for service requests

Jira Service Management allows you to customize the fields of information collected from customers. Additionally, you can customize the fields of information used by your agents. Jira Service Management does this through issue type fields and screens. Fields help agents fulfill the request, discuss with vendors, and categorize requests.

By default, we include the following fields in your agents' view of a service request:

Field name	Description
Summary	A short description of the request.
Reporter	The person who submitted the request.
Component/s	Segments of your IT infrastructure that relate to the request. For example, "Billing services" or "VPN server". These are used for labeling, categorization, and reporting.
Attachment	Files or images added to the request.
Description	A long, detailed description of the request.
Linked Issues	A list of other requests that affect or are effected by the request. If your business uses other Atlassian products, this list may include linked development issues.
Assignee	The service desk agent assigned to fulfill the request.
Priority	The importance of the request's resolution, usually in regards to your business needs and goals. Sometimes, priority is calculated by impact and urgency.
Labels	A list of additional custom labels used for categorizing or querying records.
Request participants	A list of extra customers or vendors who take part in resolving the request.
Approvers	A list of business or financial contacts responsible for approving the service request.
Organizations	A list of customer or vendor groups interested in the request's resolution.

Extra form fields recommended by ITIL

ITIL recommends a few more fields for their in-depth processes. The IT Service Desk template doesn't include these by default. This is because IT teams who use Jira Service Management don't often use these fields. If needed, you can include these fields or add custom fields. [Find out more about fields in Jira.](#)

ITIL also recommends including the following fields:

Field name	Description
Impact	The effect of the service request, usually in regards to service level agreements.
Urgency	The time available before the business feels the service request's impact.
Pending reason	A short description or code that indicates why the service request is not progressing.
Product categorization	A category of IT asset or system that the request effects.

Operational categorization	A category of action or function required to fulfill the request.
----------------------------	---

Learn more about IT service management (ITSM)

Get more tips and tricks for successful [ITSM](#), view case studies, and learn how to take your service desk to the next level.

[Check out the ITSM resources on IT Unplugged.](#)

Managing changes with your IT service project

Effective service teams plan and control changes, as they understand their impact to their business. Although Jira Service Management provides default change management workflows and fields, we've now updated them to better follow the Information Technology Infrastructure Library (ITIL) best practices. We've also added a few additional steps that will let you create an inventory of assets and open changes against them by using the built-in asset management functionality.

Learn more about ITIL and change management

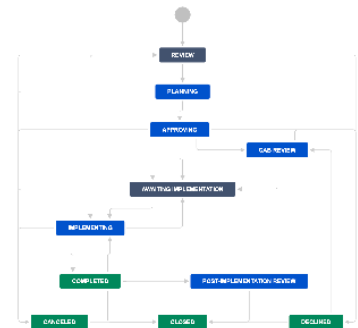
If you're not familiar with the ITIL best practices, check out our resources. They'll explain you the basic concepts, definitions, and processes. [Learn more about change management](#)

Set up change management in Jira Service Management

Have a look at the list of customizations we've made to the change management process. If you'd like to use them, you'll need to adjust your individual projects manually.

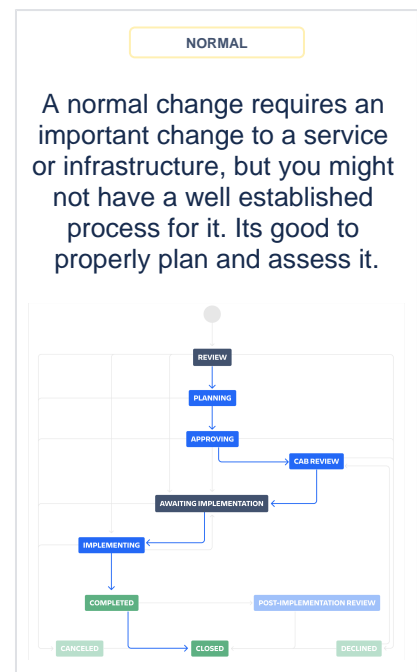
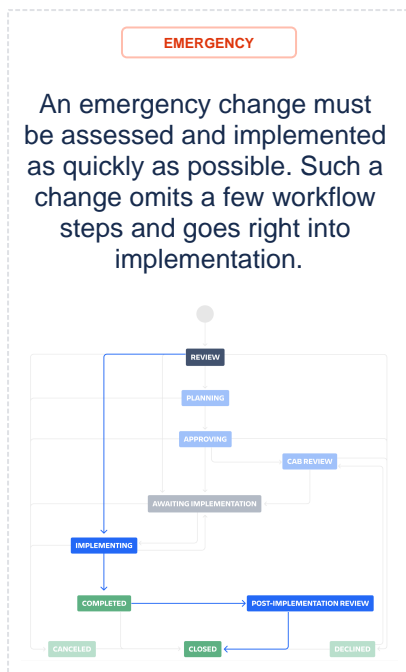
Change management workflow

The workflow follows the ITIL best practices and terminology, and makes sure your changes are approved and accounted for. Extra steps let you properly plan and assess the most important and risky changes, at the same time not slowing down critical changes that must be implemented right away. All this with the right stakeholders sharing their expertise and approving your changes along the way.



Different types of changes using different workflow paths

Thanks to automation rules, different types of changes can take different paths. Just so emergency changes don't have to wait for all of the approvals they're emergencies after all.



Automation rules handling repetitive tasks for you

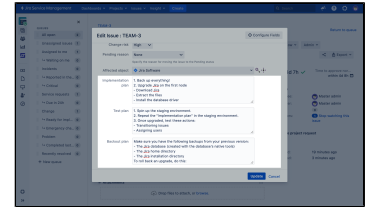
Automation rules will also take some burden off your back and automate repetitive tasks, just so you dont have to think about them:

- Determine change risk based on the importance of affected assets
- Calculate priority based on Impact and Urgency
- Escalate Emergency changes
- Auto-approve Standard changes

Change plans

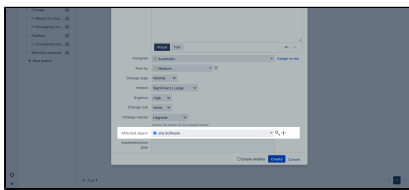
With custom fields, each change requests can store detailed plans, just to make sure everyone working on it knows what theyre doing or can quickly roll back if it turns out they dont.

- Implementation plan
- Test plan
- Backout plan

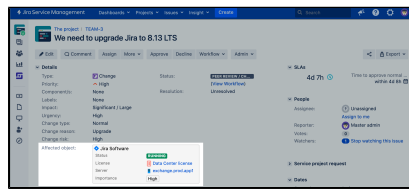


Inventory of assets and configuration items

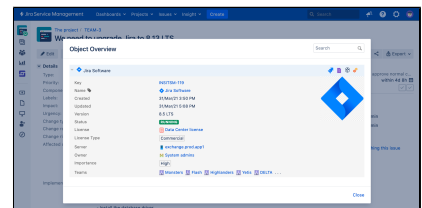
Jira Service Management lets you create an inventory of assets and configuration items, together with their details and relations between them. You can then open changes against these assets and get them linked together. Thanks to that, details about an asset are included in the change request and everyone involved has enough context to decide how risky or important a change is.



Selecting an asset



Viewing an asset

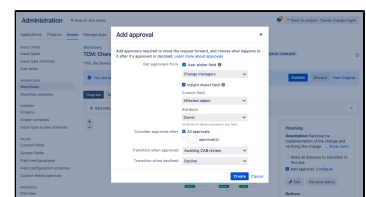


Viewing more details

Approvals from the right stakeholders

Make sure the most important and risky changes are approved by the right people. You can include change managers, CAB, and approvers that are directly responsible for your assets the right people at the right time. By using Insight custom fields, these approvers will change dynamically depending on the asset affected by your change request.

- Change managers (user picker field)
- Change Advisory Board (user picker field)
- Owners of your Insight assets (Insight field)



Get started with improved change management

To get started, follow these steps. Theyll help you adjust your projects to all of these changes and provide some explanations about what these changes do:

- [What to expect: Sample lifecycle of a change](#)
- [1. Update the change management workflow](#)
- [2. Create resolutions, post-functions, and priorities](#)
- [3. Create and update custom fields](#)
- [4. Link your project to Insight asset management](#)
- [5. Add approvals to your workflow](#)

- [6. Create and update automation rules](#)
- [7. \(Optional\) Set up a calendar to coordinate your changes](#)

What to expect: Sample lifecycle of a change

Here's an example of how a change request could look like, and how it progresses through the change management workflow. This should help you understand the changes you'll be making to your projects.

Requesting a software upgrade Jira Software


This example shows a request for a software upgrade of our Jira Software instance that lives as an asset in Jira Service Management.

What we're doing	What it looks like in Jira
<p>Viewing an asset in Insight</p> <p>Lets have a look at our Jira instance (asset) in Insight. Its quite an old version and we want to get it upgraded to get new features, security updates, and all the other bug fixes.</p>	

Well open a change request against it to upgrade it to the latest LTS release : 8.13.

Jira Service Management Dashboards ▾ Pro

Applications / Jira Software








INSITSM-119

Jira Software

📎 Attachment

✍️ Comment

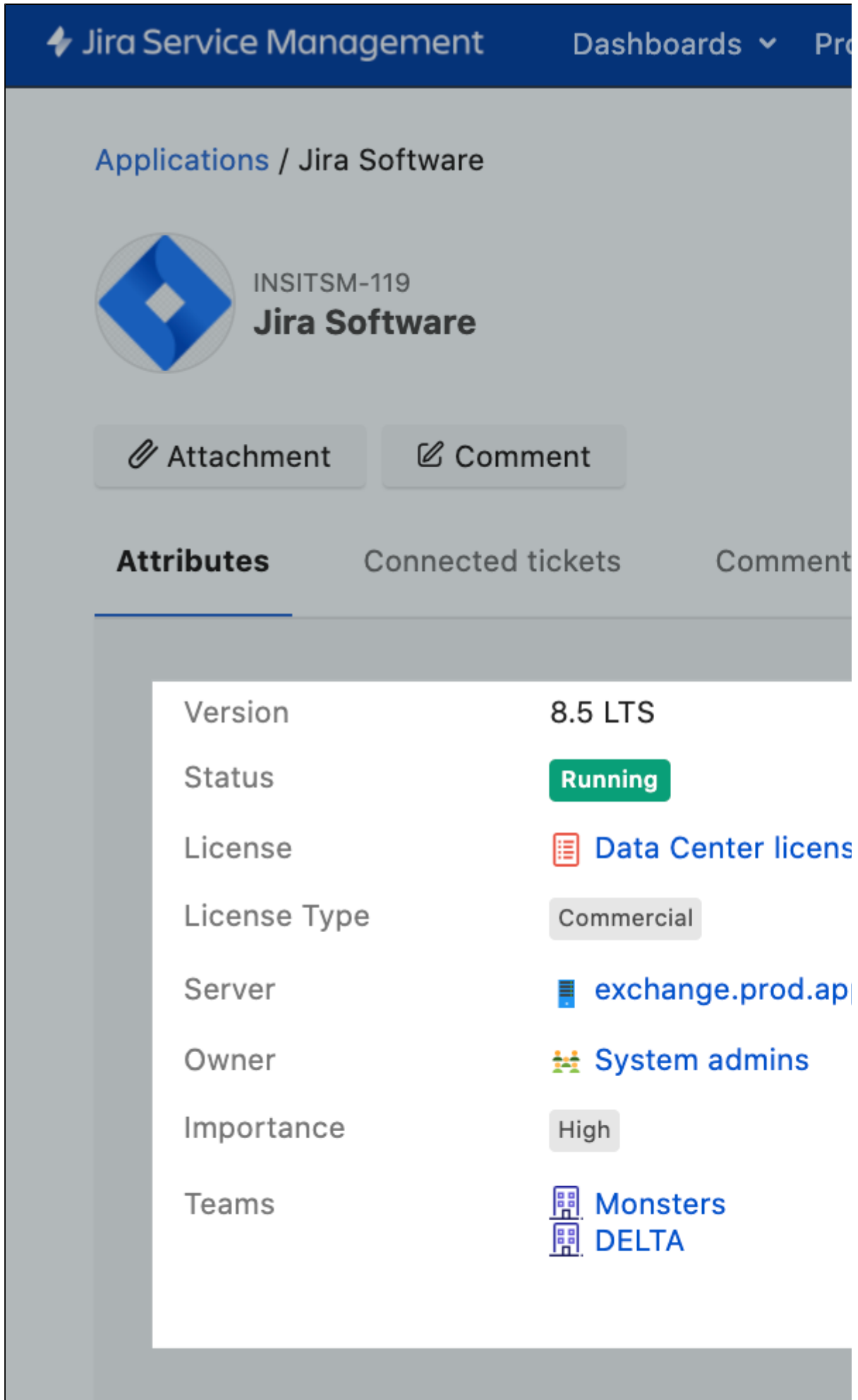
Attributes
Connected tickets
Comment

Version	8.5 LTS
Status	Running
License	 Data Center licens
License Type	Commercial
Server	 exchange.prod.ap
Owner	 System admins
Importance	High
Teams	 Monsters  DELTA

Whats important here is some of the fields of our asset that will affect how our change request behaves:


- **Importance** (based on which our change will be assigned a proper risk)



- **Owner** (who needs to give a green light, because she's the system administrator responsible for this Jira instance)








Jira Service Management Dashboards ▾ Pro

Applications / Jira Software

 **INSITSM-119**
Jira Software

 Attachment  Comment

Attributes Connected tickets Comment

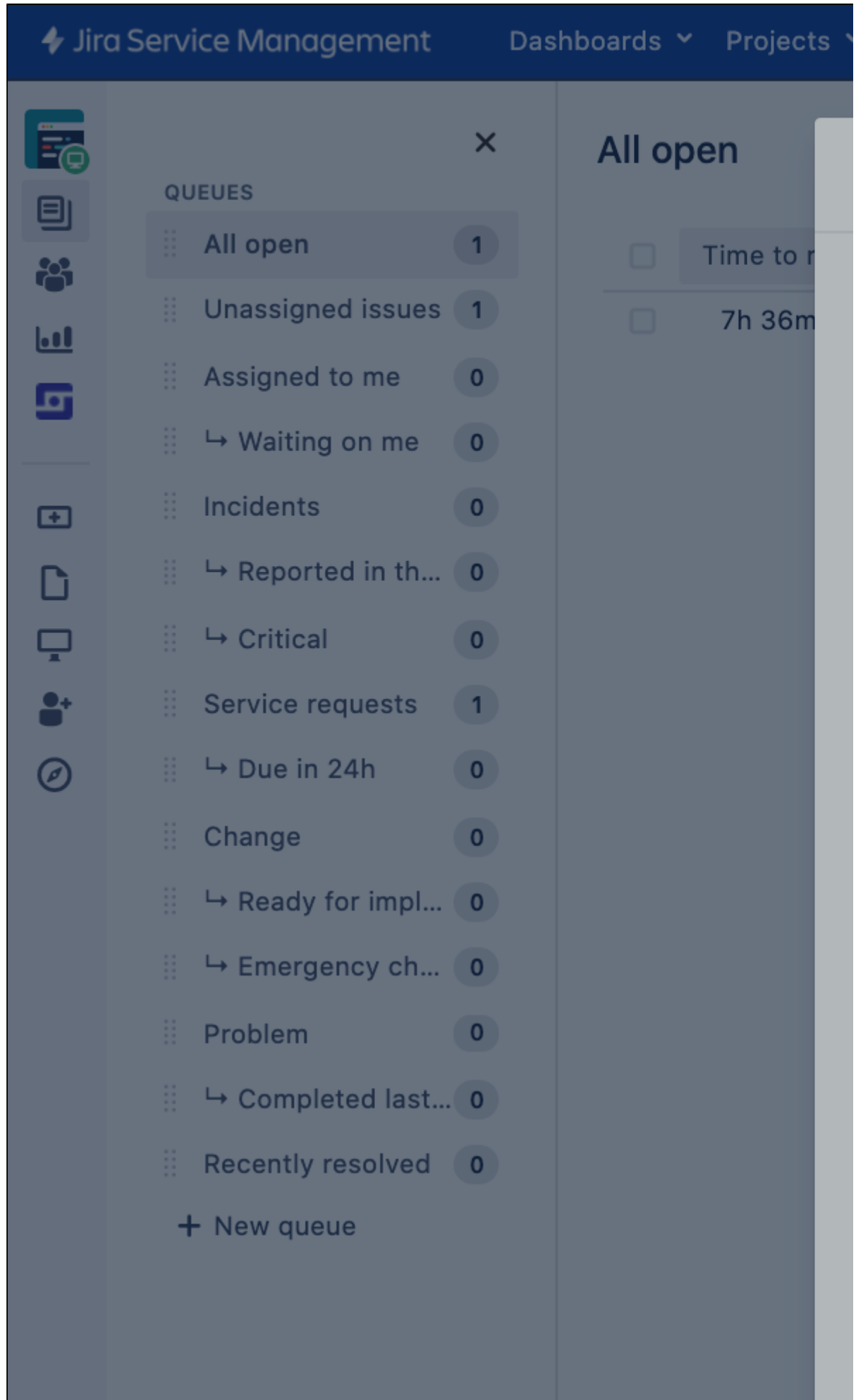
Version	8.5 LTS
Status	Running
License	 Data Center licens
License Type	Commercial
Server	 exchange.prod.ap
Owner	 System admins
Importance	High
Teams	 Monsters  DELTA

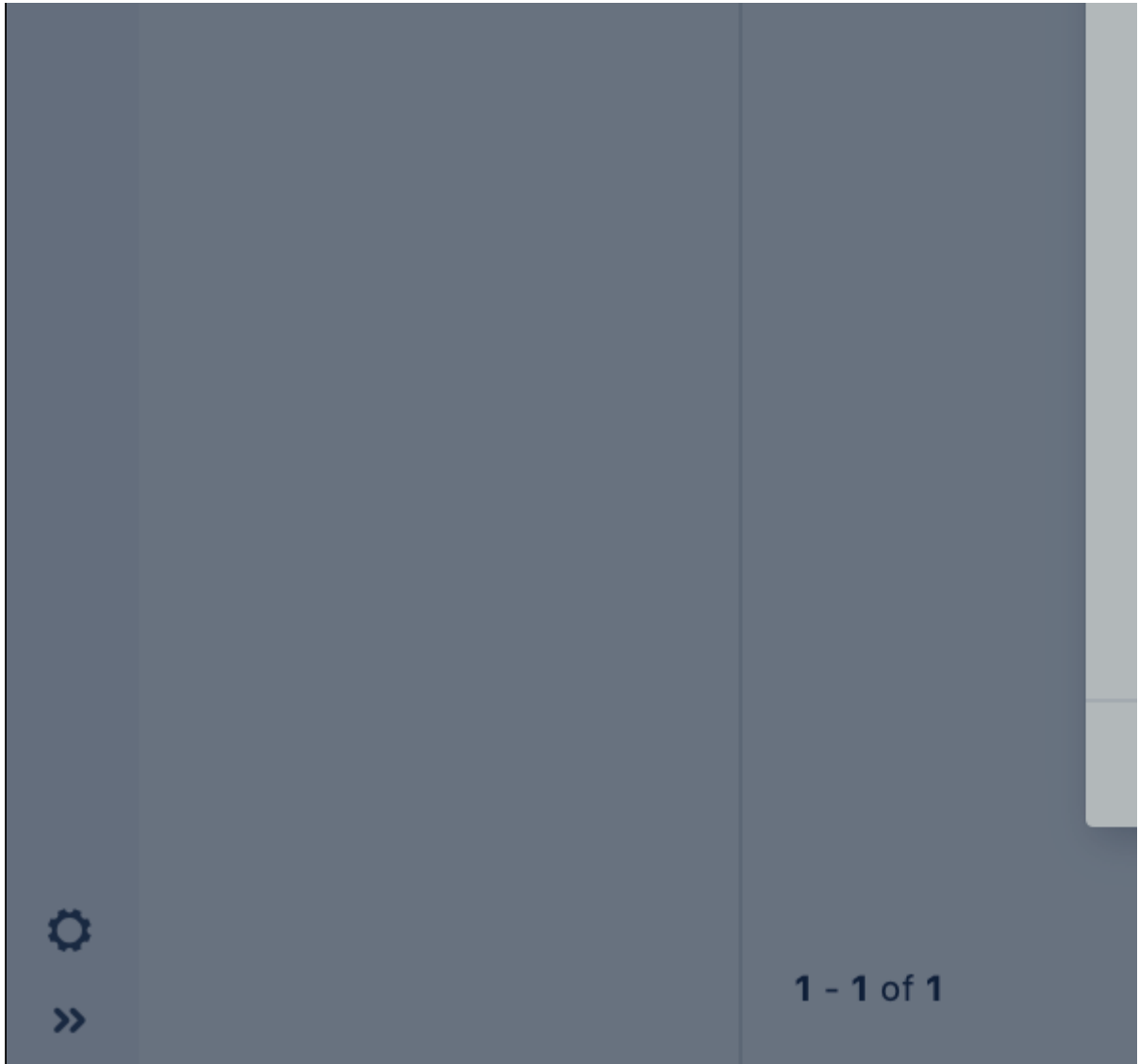
- **References to other assets:** Server, License, Teams using it (just to see dependencies)



Creating a change request

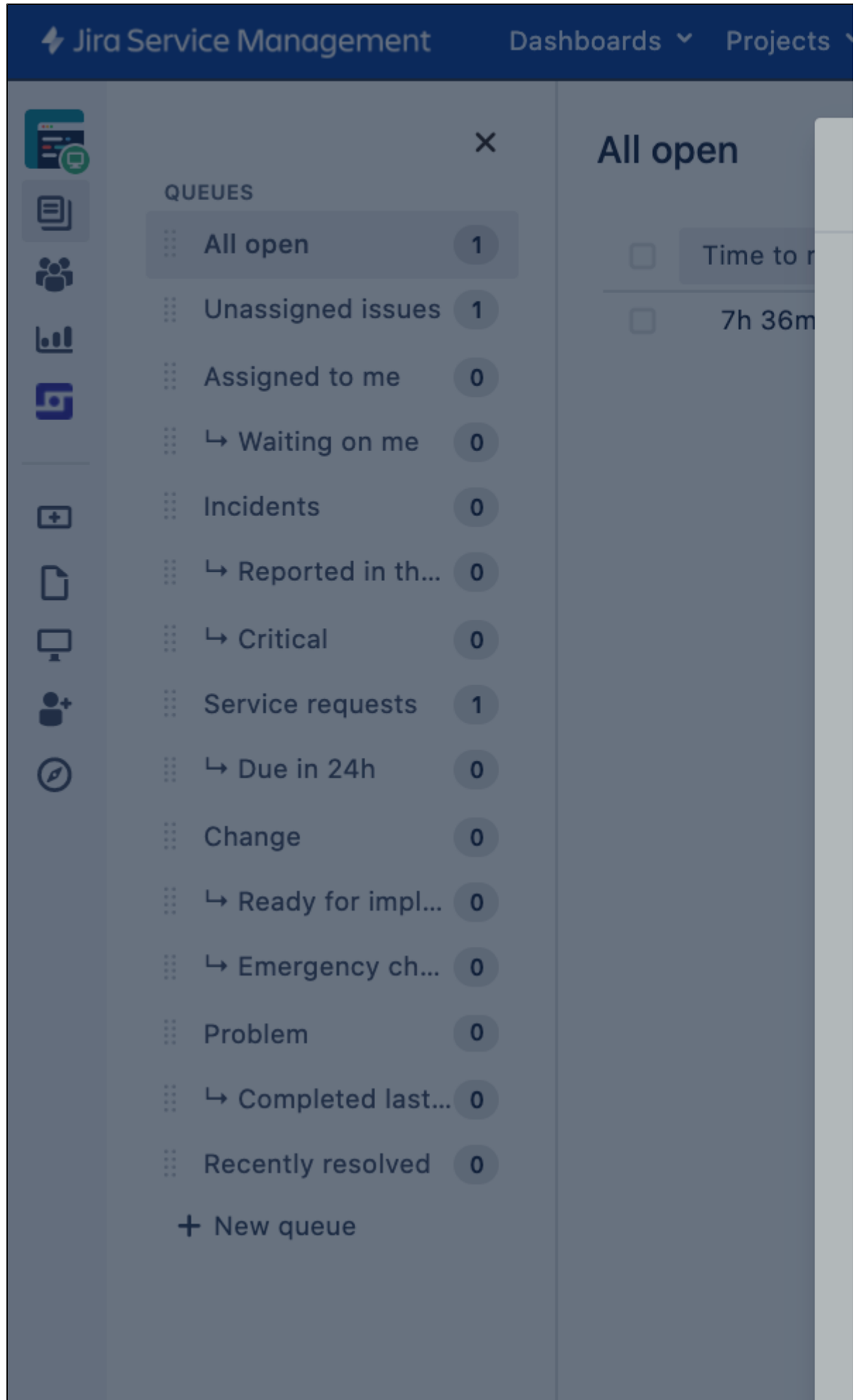
Lets open a change request, and choose this Jira instance as an affected object.

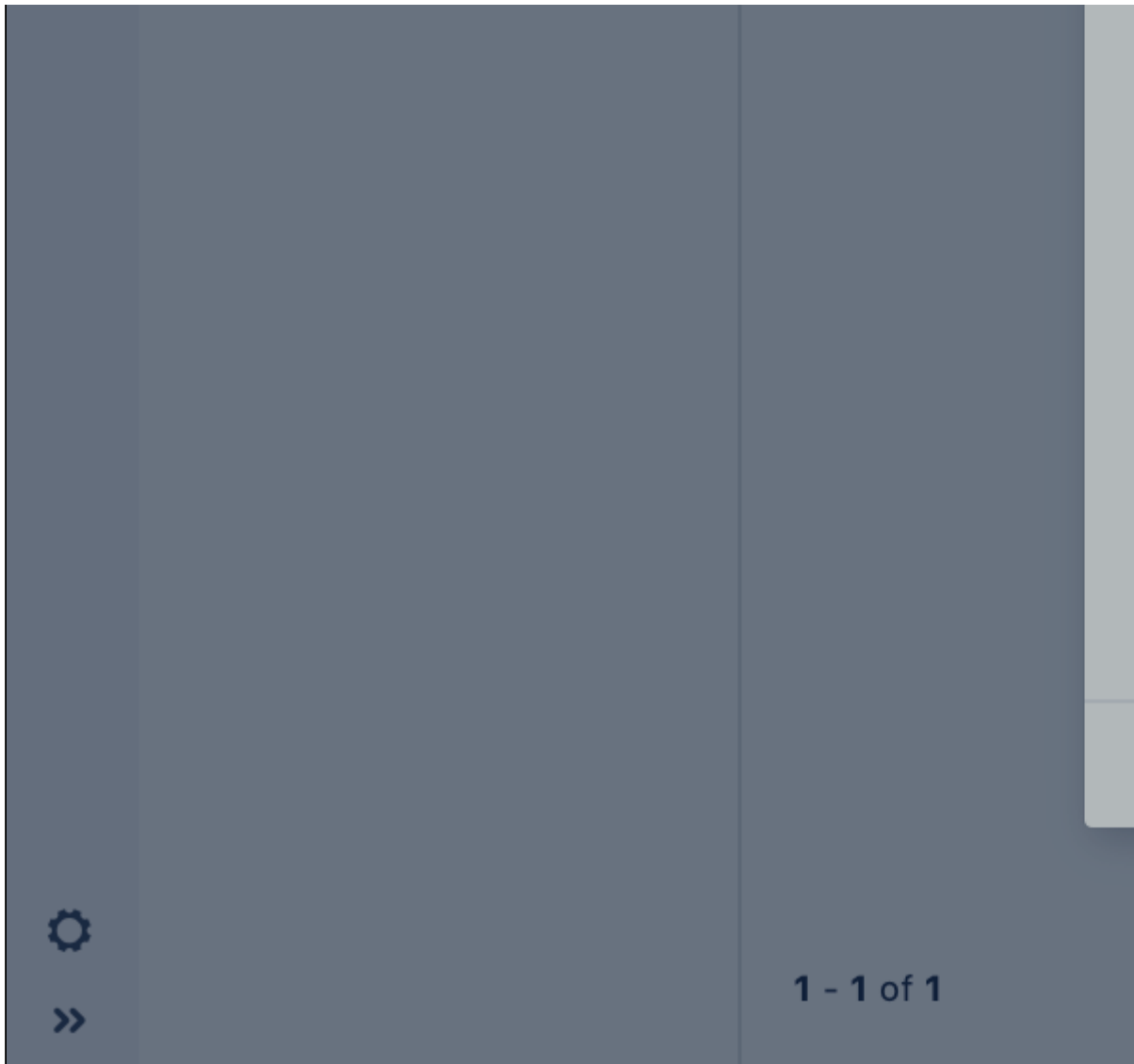




Lets fill in the fields explaining the change:

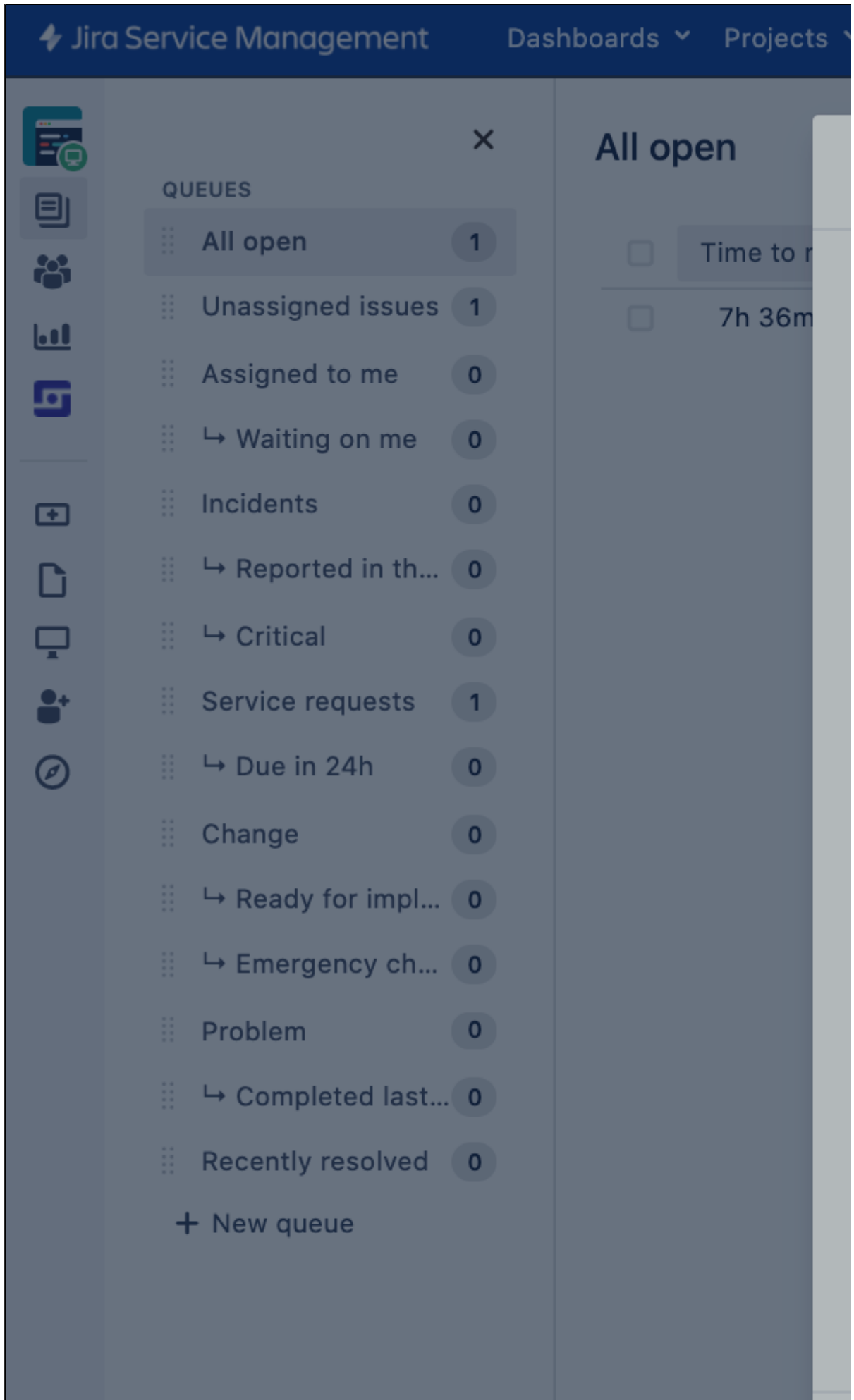
- Change type
- Change reason
- Urgency
- Impact

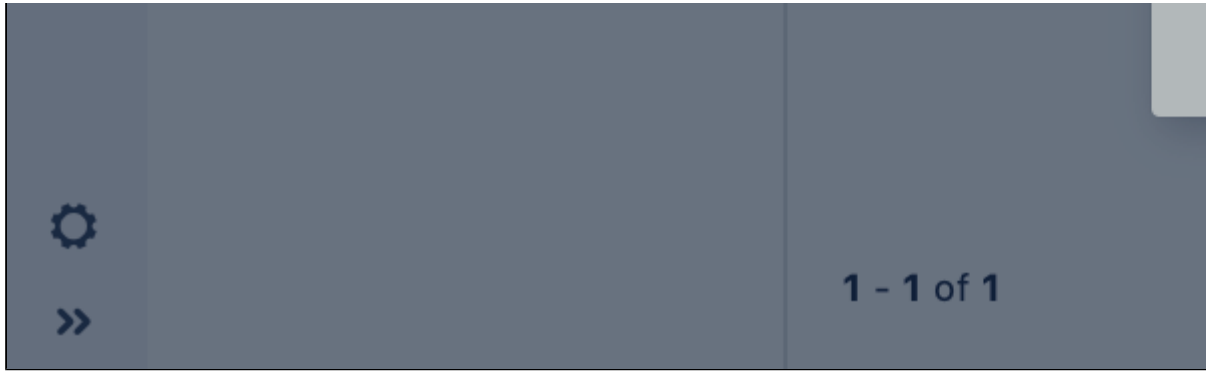




Lets
leave
the
new
fields
for
plans
empty
right
now.
Who
will
be
filling
them
out
depen
ds on
the
compa
ny
proces
susuall
y its
the
change
reques

tor, but lets try including the whole team in it later.





**Viewin
g the
asset
on the
chang
e
request**

After the change request is created, details of our Jira instance are pulled into the request. Both you, and everyone interested, can view it.

You can configure what exactly is displayed here

The screenshot shows the Jira Service Management interface. At the top, there is a blue header with the text 'Jira Service Management' and 'Dashboard'. Below the header, on the left, is a vertical sidebar with several icons representing different functions like home, search, users, reports, and settings. The main content area displays a change request card. The card has a title 'The project / TEAM-3' and a main heading 'We need to upgrade'. Below the heading are three buttons: 'Edit', 'Comment', and 'Assign'. A 'Details' section is expanded, showing various attributes of the change request: Type (Change), Priority (High), Component/s (None), Labels (None), Impact (Significant /), Urgency (High), Change type (Normal), Change reason (Upgrade), and Change risk (High). At the bottom of the details section, there is a dropdown menu for 'Affected object:' which is currently open, showing options: 'Jira Software', 'Status', 'License', 'Server', and 'Importance'.

The screenshot shows the Jira Service Management interface. At the top, there is a dark blue header with the text "Jira Service Management" and "Dashboard". Below the header, the breadcrumb "Applications / Jira Software" is visible. A large blue diamond icon represents the application. To its right, the text "INSITSM-119" and "Jira Software" is displayed. Below this, there are two buttons: "Attachment" and "Comment". A tabbed interface shows "Attributes" and "Connected tickets" (which is selected). Under the "Connected tickets" tab, there is a toggle for "Unresolved" (which is turned off) and a search box labeled "Jira filter...". A ticket card is visible, showing a purple icon, the text "TEAM-3", and the description "We need to upgrade Jira to".



**Viewin
g the
effect
of
autom
ation
rules**

But,
theres
more.
Thanks
to
autom
ation
rules,
Jira
autom
atically:

- Def
ine
d the
pri
orit
y as
High
- Def
ine
d the
cha
ng
e
risk
as
High

This will help the approvers take necessary steps when approving this change for example, they could move the date to a weekend and make sure that the teams using this Jira instance can work without it.

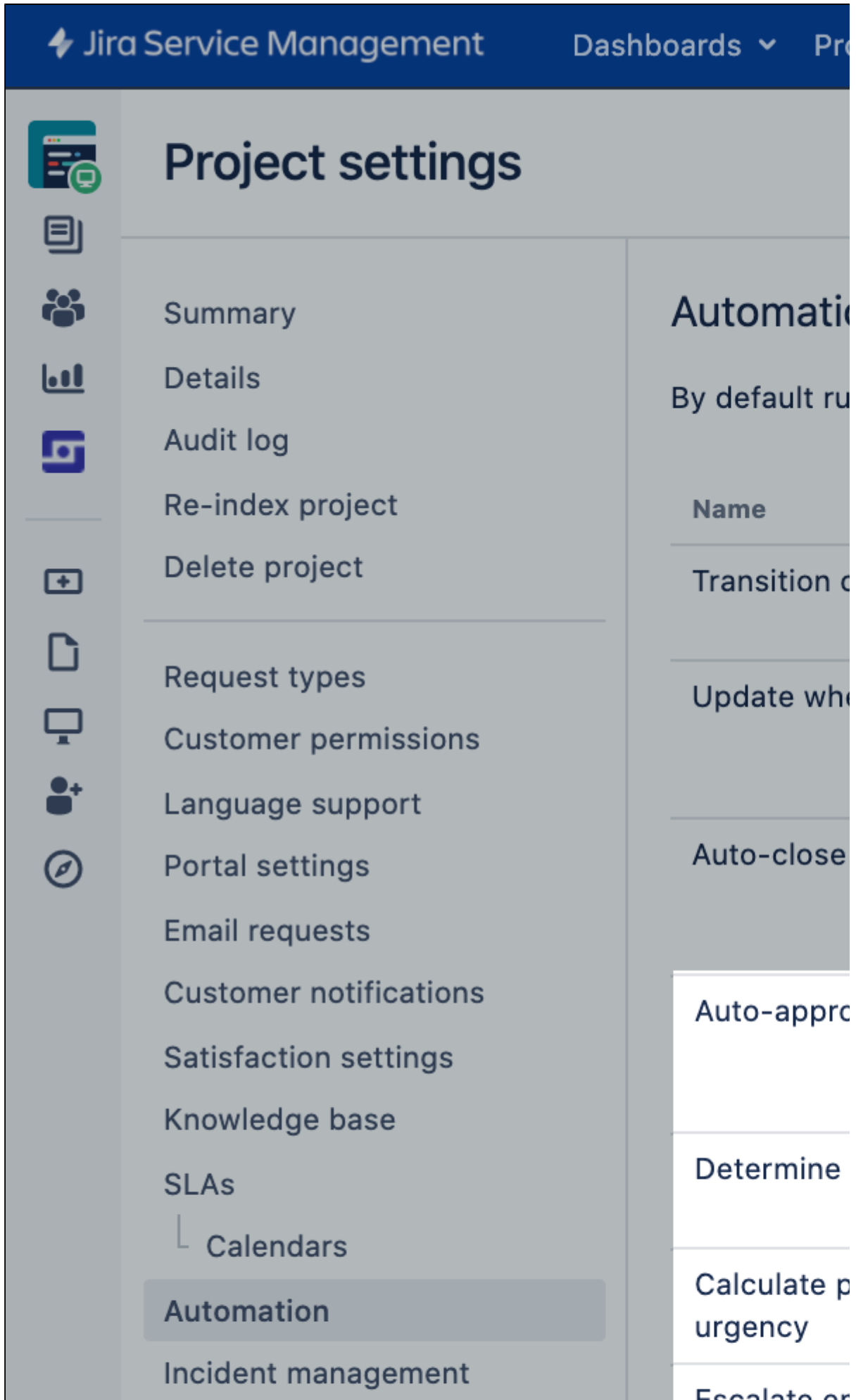
The screenshot shows the Jira Service Management interface. At the top, there's a blue header with the Jira logo and the text 'Jira Service Management' and 'Dashboard'. Below the header, there's a sidebar with various icons. The main content area shows a change request for 'The project / TEAM-3' with the title 'We need to upgrade...'. There are three buttons: 'Edit', 'Comment', and 'Assign'. Below these is a 'Details' section with a dropdown arrow. The details include: Type: Change, Priority: High, Component/s: None, Labels: None, Impact: Significant / High, Urgency: High, Change type: Normal, Change reason: Upgrade, Change risk: High, and Affected object: Jira Software. A dropdown menu is open for 'Affected object', showing options: Jira Software, Status, License, Server, and Importance.

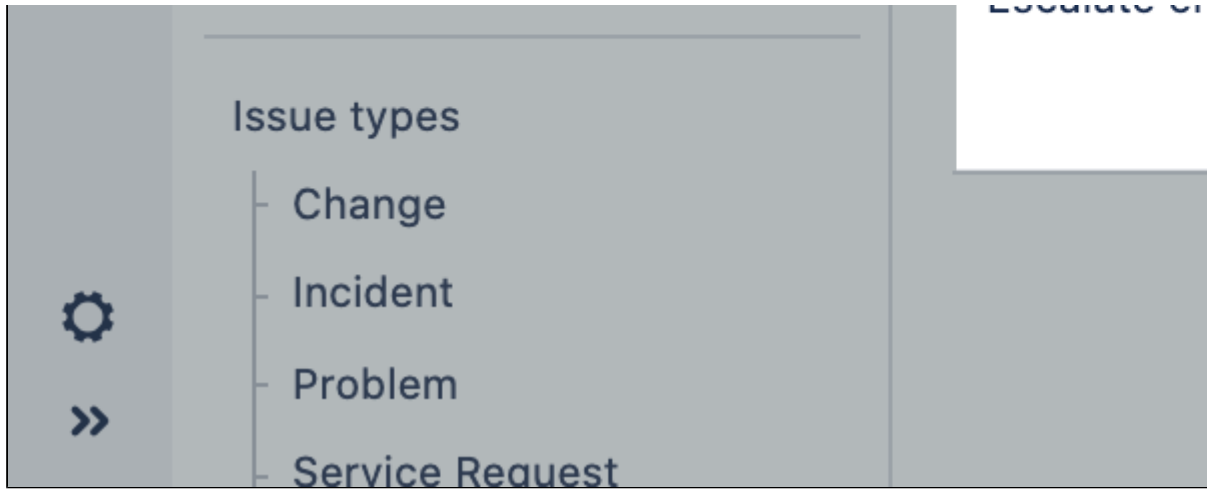


How did we do it?

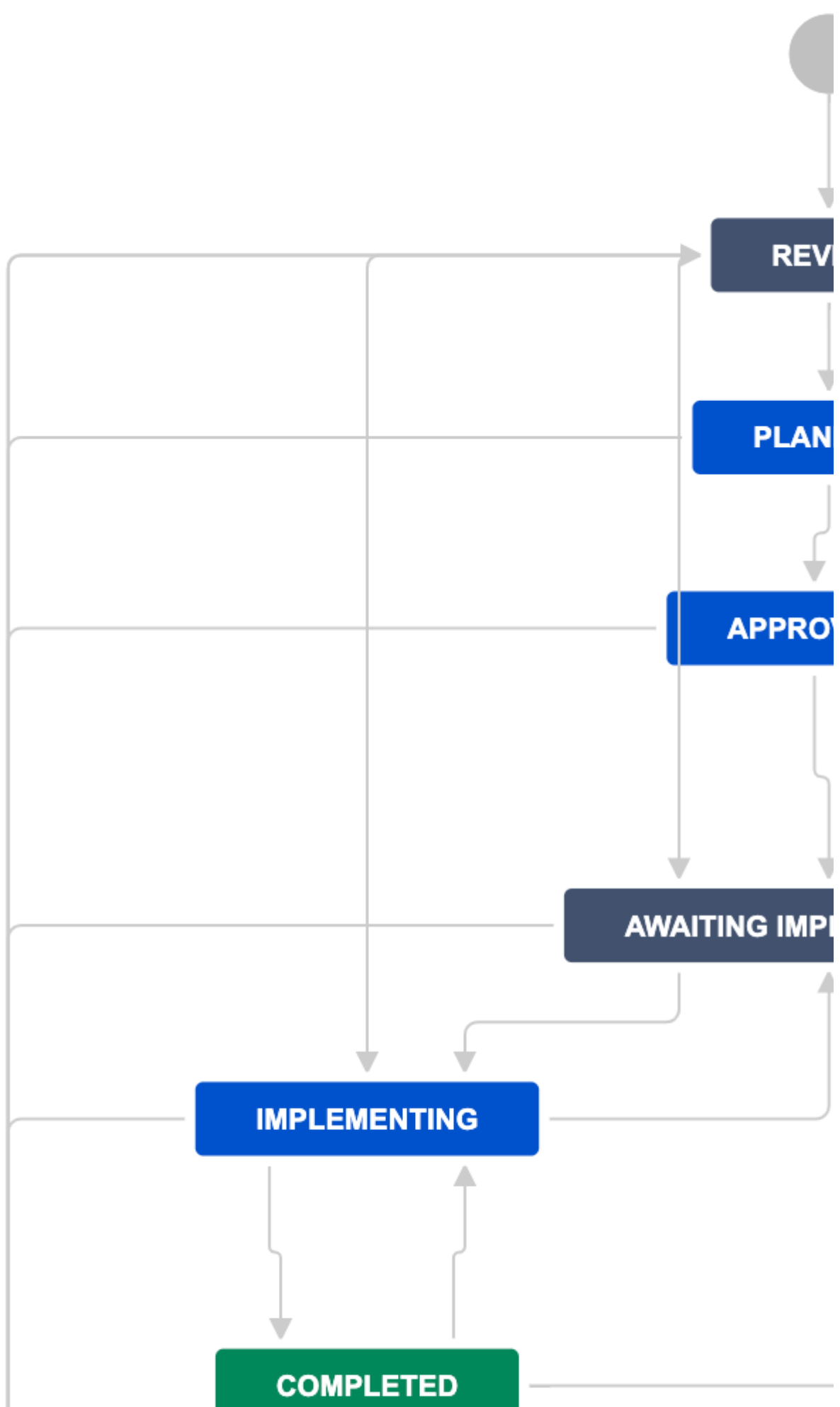
We added new automation rules to Jira Service Management. The ones used here:

- Use assets importance to determine the change risk
- Use change requests impact/urgency to assign a priority





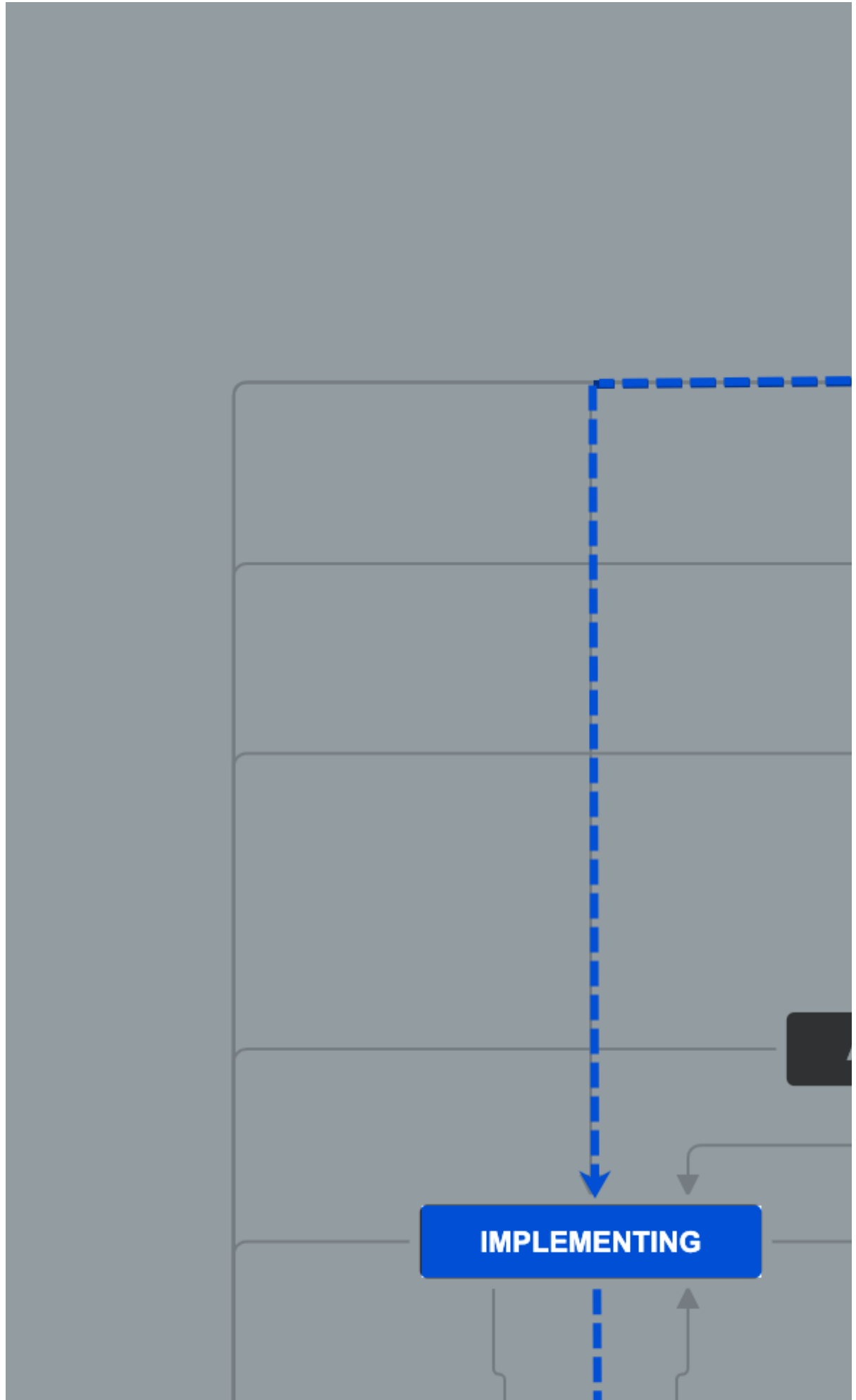
In this case, automation rules don't affect the change request that much, because *Normal* changes should go through all workflow steps and approvals they usually don't have established processes, yet are not emergencies, so we have time to review them.

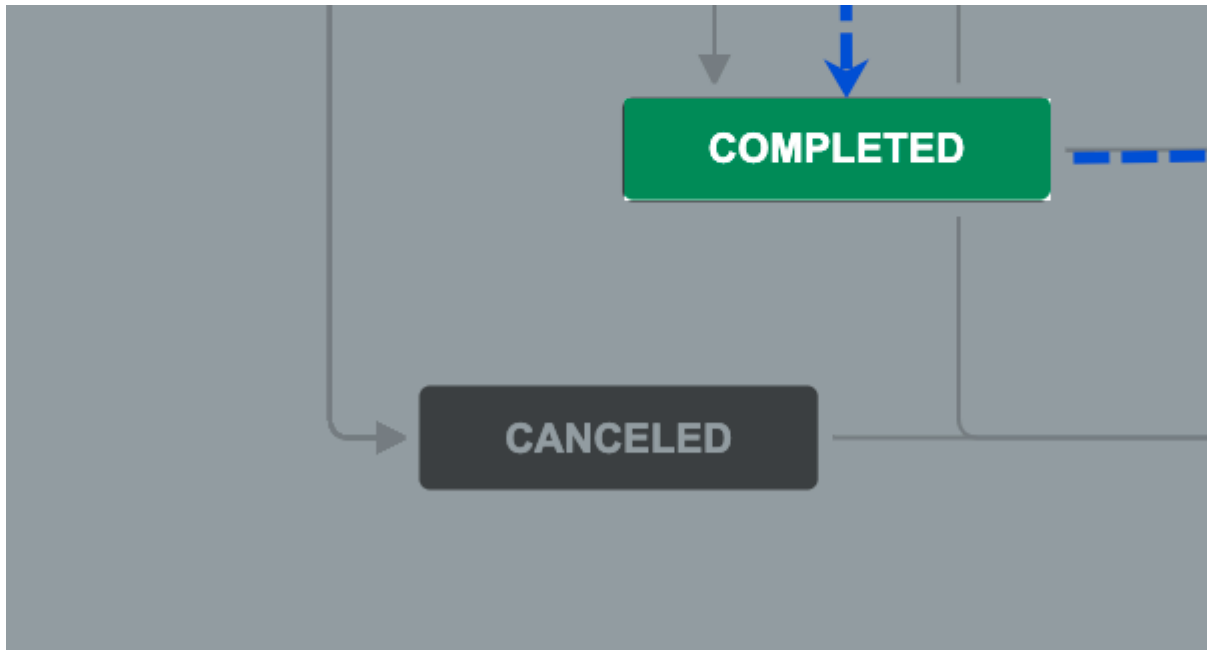




But if it was an Emergency that's a different story. Thanks to automations, Emergency changes can bypass a few steps, so you can implement them right away instead of running around asking for permissions.

It's an Emergency after all.



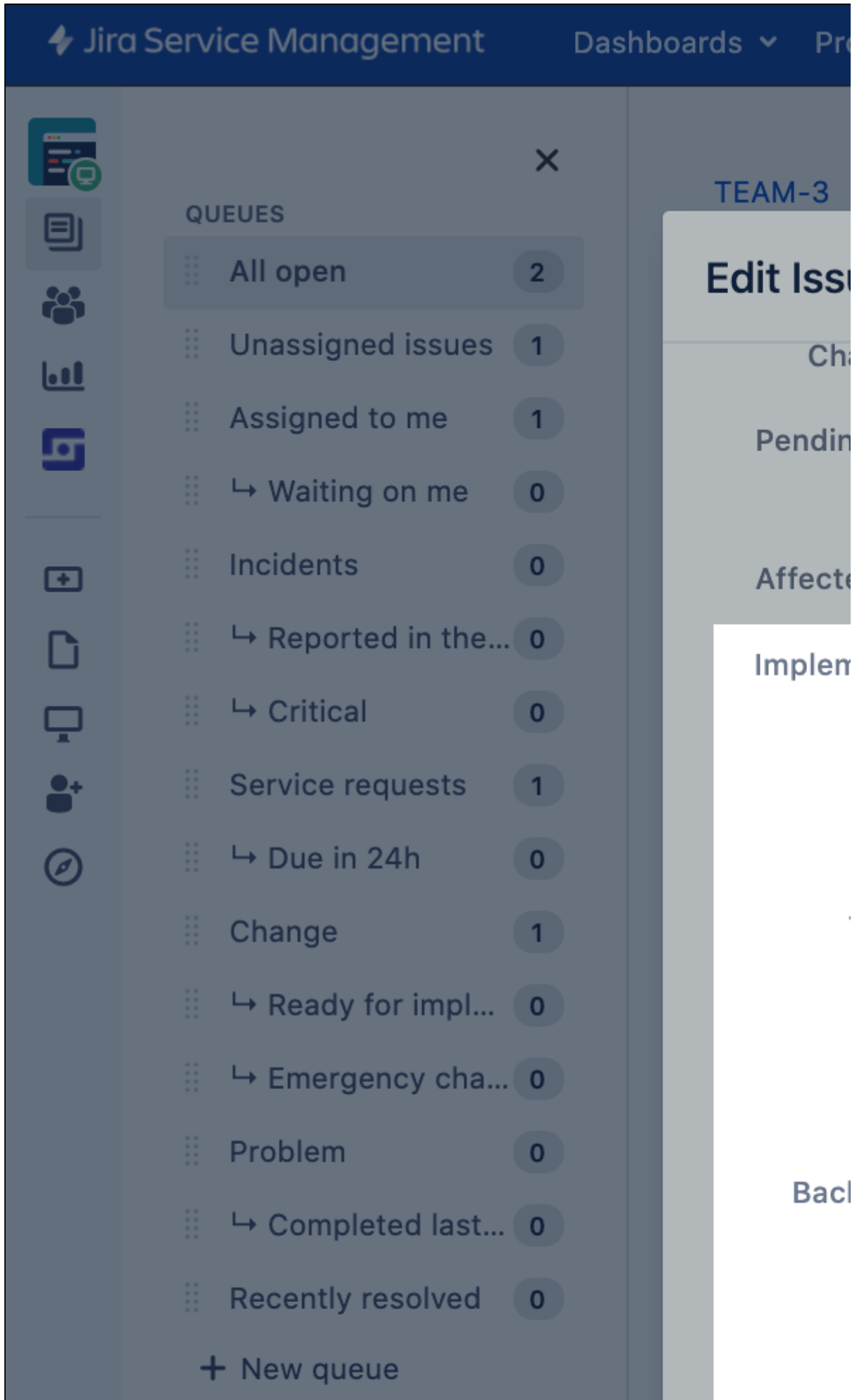


**Planni
ng the
change**

After our change request is transitioned to the Planning stage, the whole team can get together and collect all the necessary info.

- Implementation plan (how will we do it?)
- Test plan (how will we test it works?)
- Backout plan (how do we quickly roll back?)

Just to make sure we know what were doing or can roll back if it turns out we dont.





Approving the change by the assets owner

With the plan ready, we can transition on the change request to the Approving step where it will undergo the initial review. It would usually be done by change managers (fetched from a Jira user picker custom field) and someone responsible for this Jira instance the right person at the right time.

The screenshot shows a Jira Service Management interface. At the top, there's a navigation bar with 'Jira Service Management', 'Dashboards', and 'Pro'. Below this is a sidebar with various icons. The main content area is titled 'TEAM-3' and features a large heading: 'We need to upgrade Jira to 8'. Below the heading are buttons for 'Edit', 'Comment', 'Assign', and 'More'. A 'Details' section is expanded, showing the following information:

- Type: Change
- Priority: High
- Component/s: None
- Labels: None
- Impact: Significant / Large
- Urgency: High
- Change type: Normal
- Change reason: Upgrade
- Change risk: High
- Affected object: Jira Software
 - Status
 - License
 - Server
 - Importance
- Implementation plan:
 - 1. Back up every
 - 2. Upgrade Jira
 - Download Jira
 - Extract the file
 - Install the data



Although you usually know who change managers are, you can have as many asset owners as you have assets. You don't have to know who they are, because Jira will automatically assign them as approvers based on their relation to the asset.

It will be a different owner for a different software or server, or maybe no owner if the asset isn't that important.

The screenshot shows the Jira Service Management interface for an application. At the top, there is a navigation bar with 'Jira Service Management', 'Dashboards', and 'Pr'. Below this, the breadcrumb 'Applications / Jira Software' is visible. The main content area features the application icon (a blue diamond) and the identifier 'INSITSM-119 Jira Software'. There are two buttons: 'Attachment' and 'Comment'. Below these are three tabs: 'Attributes', 'Connected tickets', and 'Comments'. The 'Attributes' tab is active, showing a list of key-value pairs for the application: Version (8.5 LTS), Status (Running), License (Data Center license), License Type (Commercial), Server (exchange.prod.app1), Owner (System admins), Importance (High), and Teams (Monsters, DELTA).

Attribute	Value
Version	8.5 LTS
Status	Running
License	Data Center license
License Type	Commercial
Server	exchange.prod.app1
Owner	System admins
Importance	High
Teams	Monsters, DELTA

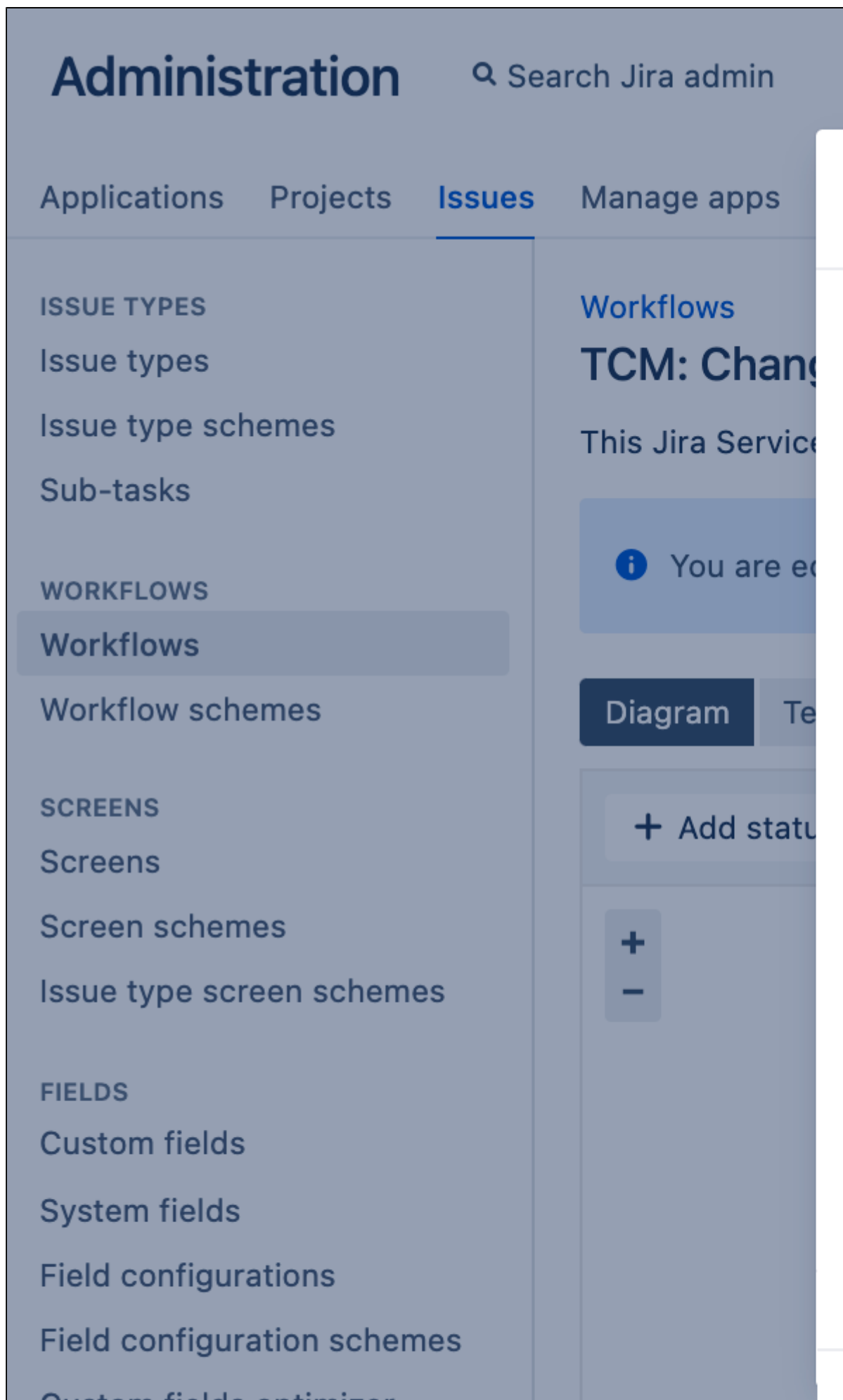


How did we do it?

We used approvers who are related to the Insight object. It wont always be a fixed group of users they will change based on the asset you select in the change request.

Jira will:

- Check what the affected asset
- Look for approvals in its attributes
- Assign the separate approvers





**Asses
sing
the
risk:
CAB
review**

After the initial review, the change is moved to the next approval step. Normal changes don't have established processes so it's good to properly assess them.

Here the change is being assessed by the Change Advisory Board (CAB) that can include additional experts to reduce the risk and avoid any problems.

The screenshot shows a Jira Service Management interface. At the top, there's a navigation bar with 'Jira Service Management', 'Dashboards', and 'Pr'. Below this is a sidebar with various icons. The main content area displays a change record for 'TEAM-3' with the title 'We need to upgrade Jira to 8'. There are buttons for 'Edit', 'Comment', 'Assign', and 'More'. A 'Details' section is expanded, showing the following information:

Type:	Change
Priority:	High
Component/s:	None
Labels:	None
Impact:	Significant / Large
Urgency:	High
Change type:	Normal
Change reason:	Upgrade
Change risk:	High
Affected object:	Jira Software

The 'Affected object' dropdown menu is open, showing options: Status, License, Server, and Importance. Below the details, the 'Implementation plan' is visible, starting with '1. Back up every' and '2. Upgrade Jira', followed by sub-points: '- Download Jira' and '- Extract the file'.



Implementing the change

Once the change is approved, we can start implementing it. Lets assume that all the teams have been informed and we can just upgrade this instance.

The screenshot shows a Jira Service Management interface. At the top, there is a navigation bar with 'Jira Service Management', 'Dashboards', and 'Pr'. On the left, a vertical sidebar contains several icons: a dashboard, a document, a group of people, a bar chart, a Jira logo, a plus sign in a box, a document, a monitor, a person with a plus sign, and a magnifying glass. The main content area is titled 'TEAM-3' and features a large heading: 'We need to upgrade Jira to 8'. Below the heading are four buttons: 'Edit', 'Comment', 'Assign', and 'More'. A 'Details' section is expanded, showing the following information: Type: Change; Priority: High; Component/s: None; Labels: None; Impact: Significant / Large; Urgency: High; Change type: Normal; Change reason: Upgrade; Change risk: High; Affected object: Jira Software. A dropdown menu for 'Affected object' is open, listing 'Jira Software', 'Status', 'License', 'Server', and 'Importance'. At the bottom, the 'Implementation plan' is visible, starting with '1. Back up every' and '2. Upgrade Jira', followed by sub-points: '- Download Jira' and '- Extract the file'.



**Optional:
Changing the
objects
status**

But, in case someone was wondering what's going on with this Jira instance, starting the work on it triggered a status change.

- Jira is no longer **Running**, as we're implementing the change. Its **In Service**.

This can be done through Insight post-functions added to the workflow. In this case, they change the objects attribute when the change request is moved through a certain transition.

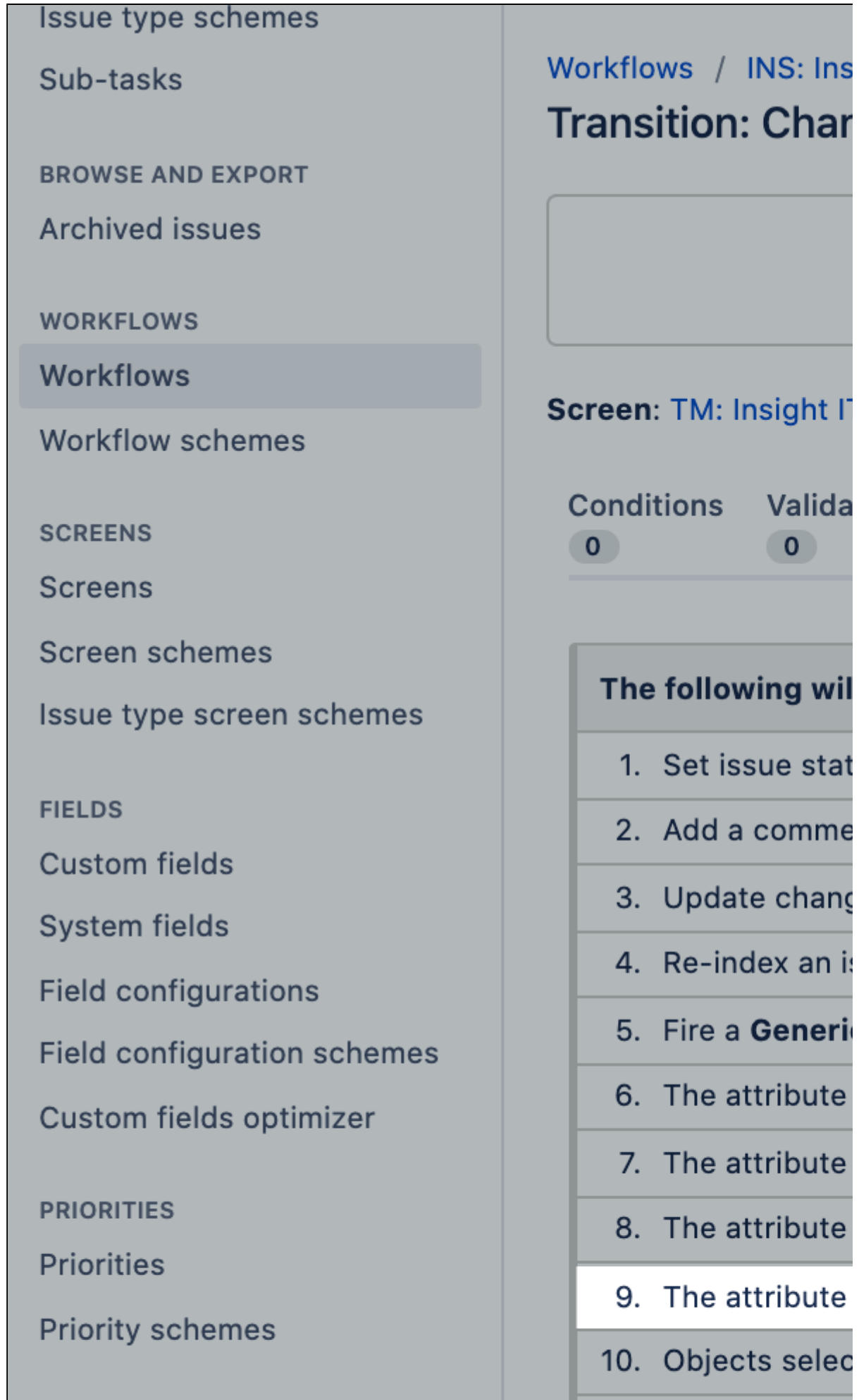
The screenshot shows the Jira Service Management interface. At the top, there's a navigation bar with 'Jira Service Management', 'Dashboards', and 'Pro'. Below this is a sidebar with various icons. The main content area displays a change request for 'TEAM-3' with the title 'We need to upgrade Jira to 8'. There are buttons for 'Edit', 'Comment', 'Assign', and 'More'. The 'Details' section is expanded, showing the following information:

- Type: Change
- Priority: High
- Component/s: None
- Labels: None
- Impact: Significant / Large
- Urgency: High
- Change type: Normal
- Change reason: Upgrade
- Change risk: High
- Affected object: Jira Software

The 'Affected object' dropdown menu is open, showing options: Status, License, Server, and Importance. Below the details, the 'Implementation plan' is visible, starting with '1. Back up every' and '2. Upgrade Jira c'.

This is not part of change management. If you'd like to use it, see [Adding Insight functions to workflows in Jira](#).







Completing and closing the change

Once Jira is upgraded and the work is done, we can complete the change, and see how everything is green again.

When closing the change request, you can select the resolution: *Change successful* or *Change failed*.

Jira Service Management Dashboards ▾ Pr

TEAM-3

We need to upgrade Jira to 8

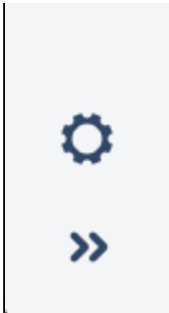
Edit
Comment
Assign
More

▾ **Details**

Type:	Change
Priority:	High
Component/s:	None
Labels:	None
Impact:	Significant / Large
Urgency:	High
Change type:	Normal
Change reason:	Upgrade
Change risk:	High
Affected object:	<div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> <p> Jira Software</p> <p>Status</p> <p>License</p> <p>Server</p> <p>Importance</p> </div>

Implementation plan:

- ▾ 1. Back up every
- 2. Upgrade Jira (
 - Download Jira
 - Extract the file
 - ...



- Install the data
- Re-apply modifi
- Disable automat
- Post-upgrade st
- Start Jira for the
- Update Jira Ser

Whats next?

Once the change is complete, you can also view some Insight reports for your assets and changes that affected them.

Jira Service Management Dashboards ▾ Pr

Report Manager

FILTERS << 1-2 of 2

- ★ My favorites
- 🔗 Shared with me
- 👤 Shared by me

Name or Description

Labels



All labels ▾

Owned By

All authors ▾

Clear filter

1-2 of 2

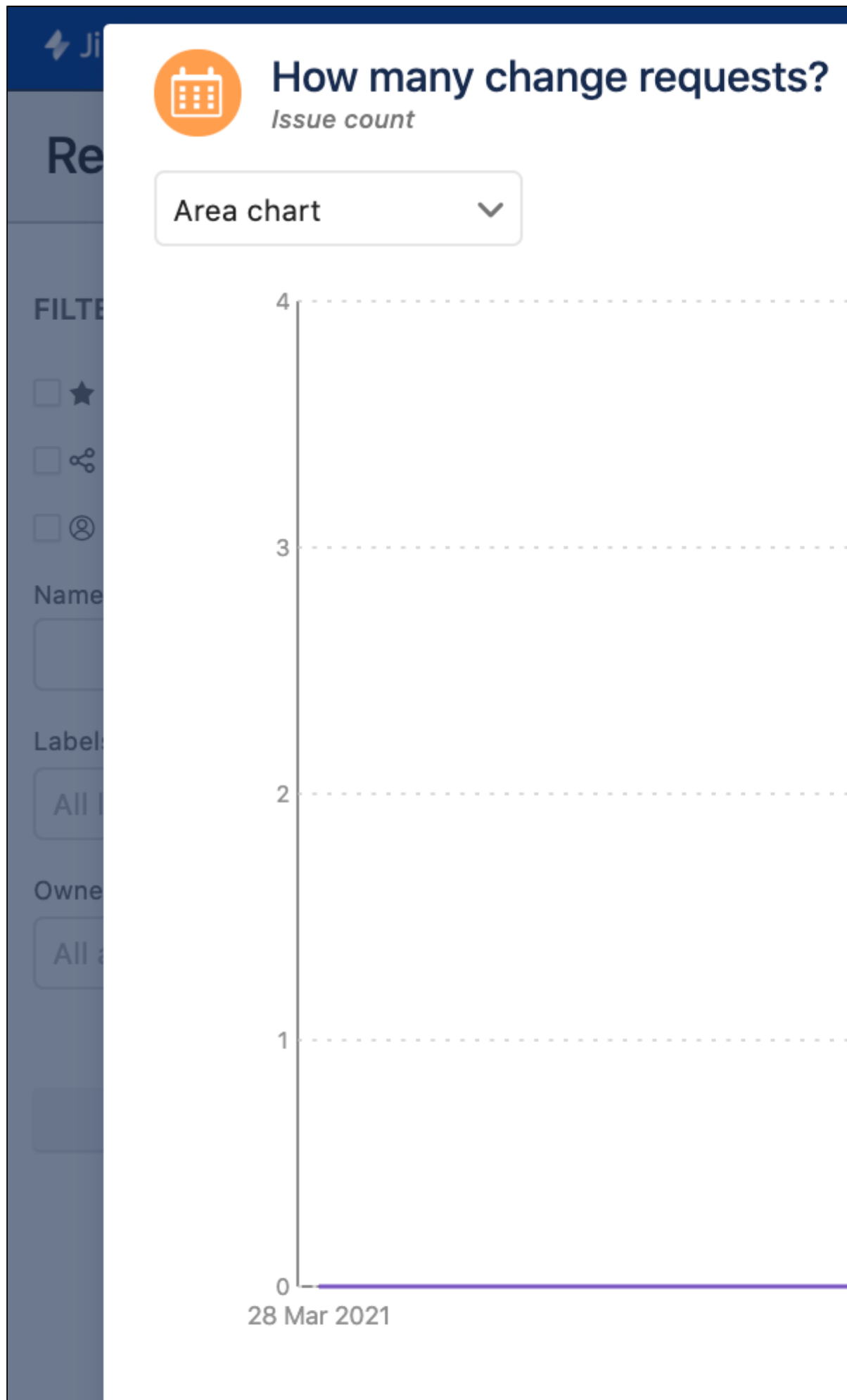
-  **How many**
Issue count
-  **Where are**
Map report

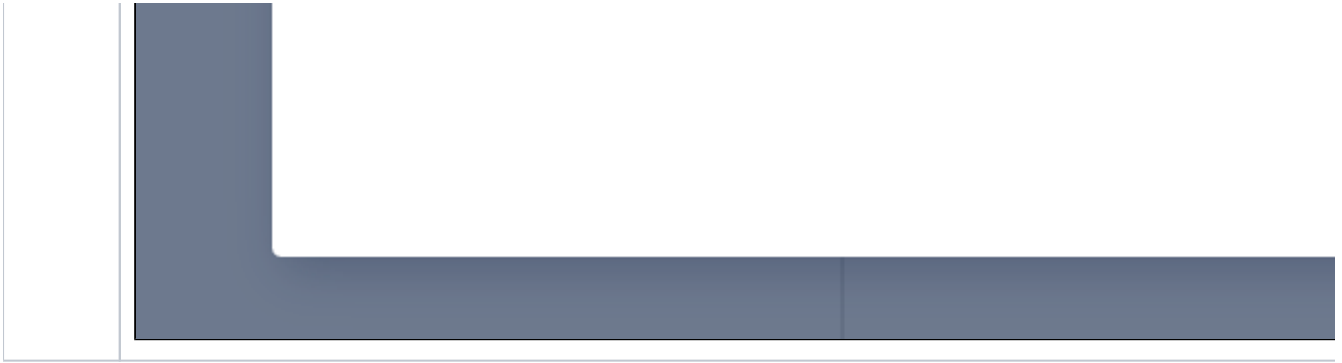
1-2 of 2



For example, you can check how many issues were raised against your assets how many incidents, problems, or changes.

And if there's too many Well, this might just mean it's time to buy a new asset.





Ready?

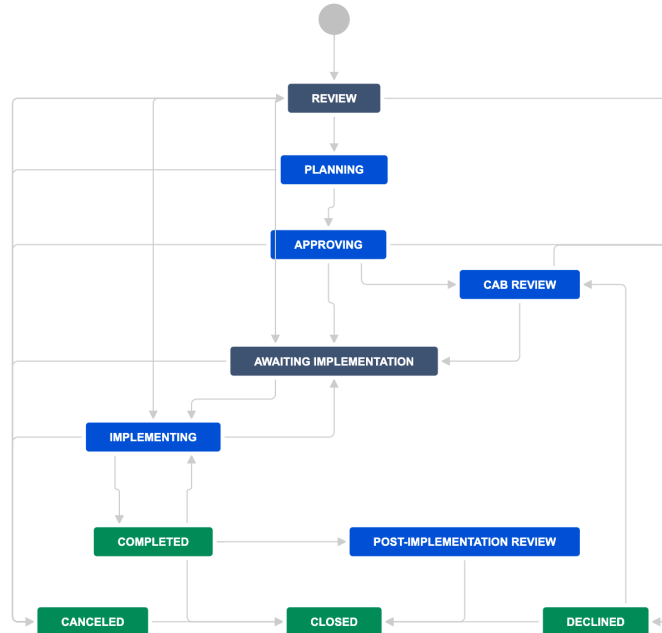
If you'd like your changes to use this flow, let's get started with [1. Update the change management workflow](#).

1. Update the change management workflow

Jira Service Management has a default change management workflow, but we've updated it with additional statuses and transitions. You can download the ready workflow from Atlassian Marketplace, and have all the missing statuses created automatically.

Your new workflow at a glance

Here's what the new workflow looks like. The statuses closely follow the ITIL best practices and terminology:



Download the workflow

To download the change management workflow:

1. Go to **Administration > Issues > Workflows**.
2. Select **Import > Import Workflow**.
3. Select **From Atlassian Marketplace**, and search for *Change management for Jira Service Management*.
4. Click **Select** and enter the workflow name.

Select a workflow


From Atlassian Marketplace From My Computer

Search for workflows Search for the workflow that you are looking for in the Atlassian Marketplace.

Change Management for Jira Service Management

A change management workflow built by the Jira Service Management team, for Jira Service Management

[Select](#) [Learn more](#)



Review statuses

After importing the workflow, Jira will automatically match the statuses to the ones that already exist in your instance and suggest creating new ones. We recommend that you do it as presented on the screen.

The following statuses would be created in a default Jira instance:

- Approving
- CAB Review
- Post-implementation review
- Review

If you're happy with them, just click **Next** and then **Import**. Alternatively, you could try mapping these missing statuses to some similar ones that you already have.

Associate the workflow with your project

To associate the workflow with your project:

1. Open your project, and go to **Project settings > Workflows**.
2. Select **Add Workflow > Add existing**.
3. Select the new *Change management for Jira Service Management* workflow.
4. Assign the *Change* issue type to this workflow.
5. When you publish the workflow, your issues will be migrated and your project will start using it.

Next steps

When you're ready, go to [2. Create resolutions, post-functions, and priorities](#).

2. Create resolutions, post-functions, and priorities

The change management workflow, and the whole process you'll be using, needs some additional configuration items, such as new resolutions, post-functions, and priorities.

Create and associate resolutions

For more info about resolutions, see [Defining resolution field values](#).

The result of the change should be reflected on the requests resolution when it transitions to *Closed*. To make it happen, create the resolutions *Changed failed* and *Change successful* and make them available when resolving a change.

To create the resolutions:

1. Go to **Administration > Issues > Resolutions**.
2. Create the following resolutions:
 - Change successful
 - Changed failed

To make the resolution available when resolving a change:

1. Go to **Administration > Issues > Workflows**, and open your change management workflow in the **Text** mode.
2. Select one of the *Close* transitions, and click **Edit**.
3. In the **Transition View** field, select your **Resolve Issue Screen**(you can use the one with this name in a default Jira instance), and save. The screen will be added to the transition and users will be able to select a resolution when closing a change through this transition.

Update Workflow Transition

i This page allows you to update the **Close** transition.

Transition Name*

Description

Destination Step

Transition View

The screen that appears for this transition (if any).

4. Repeat these steps for the remaining *Close* transitions. There are four in total.
5. Remember to publish the workflow to apply these changes.

Add post-functions to clear resolutions when the change is reopened

For more info about post functions, see [Advanced workflow configuration](#).

It might happen that you'll need to reopen a change or just move it back on the workflow. These post-functions will make sure that the resolution is cleared when that happens, just to avoid confusing your users.

To add post-functions to your workflow transitions:

1. Edit your workflow in the **Text** mode.
2. Select one of the transitions going out of the status *Closed*. Here are all of them for your reference:
 - Back to post-implementation review
 - Back to Completed
 - Back to Canceled
 - Back to Declined
3. Select **Post-functions**, and click **Add post function**. Use the following details when adding one:

- **Name:** Update Issue Field
 - **Issue field:** Resolution
 - **Field value:** None
4. Repeat the steps for the remaining transitions.
 5. Remember to publish the workflow to apply these changes.

Create the Critical priority

For more info about priorities, see [Defining priority field values](#).

Jira comes with the default Highest priority, but it doesn't have the Critical priority. Create one to follow the ITIL terminology. **It is also required** for other steps to work – automation rules that you'll configure later will use this priority.

To create the Critical priority:

1. Go to **Administration > Issues > Priorities**.
2. Select **Add priority**.
3. Call your priority **Critical**, give it some description and icon. Click **Add**.

To associate the priority with your project:

1. Go to **Administration > Issues > Priority schemes**.
2. Find the scheme used by your project and select **Edit**.
3. Move the **Critical** priority to the list of **Selected priorities**.

Edit priority scheme
USED BY 1 PROJECT

Scheme Name*
Description

Select priorities
Change the order of priorities by dragging and dropping them into the desired order. Similarly, drag and drop priorities from one list to the other to add or remove them.

Selected priorities	Available priorities
Remove all	Add all
<input type="checkbox"/> Blocker	<input type="checkbox"/> Highest
<input checked="" type="checkbox"/> Critical	<input type="checkbox"/> Lowest
<input type="checkbox"/> High	
<input type="checkbox"/> Medium	
<input type="checkbox"/> Low	
<input type="checkbox"/> Minor	

Default priority

Next steps

When you're ready, go to [3. Create and update custom fields](#).

3. Create and update custom fields

Most of the custom fields used for change management already exist in Jira so you can reuse them, but well still need to create a few theyre needed so your team has the right fields to share their plan and approach for tackling the changes.

Review existing custom fields

Here are existing custom fields that youll reuse. Were listing them for reference:

- Change type
- Change start date
- Change completion date
- Impact
- Urgency
- Change risk
- Change reason
- Change managers
- CAB
- Approvers (optional)

Update existing custom fields

Here are some recommendations on what to do with existing custom fields:

Custom field	Description
Change managers	They will be used to get your change requests approved, well do it in the next step. At this point, you can decide whether you want to include a list of approvers or let agents or customers add them manually.
CAB	<p>Here are your options:</p> <p>Agents can add or edit approvers for these fields when they fill in the details of a change request. To let customers do it, youll need to enable and show these fields on the customer portal.</p> <p>You can add approvers as default values of these fields. After the change request is created, the approvers can be modified up until the request reaches the approval steps.</p> <p>You can pull a list of approvers for these fields automatically only on the customer portal. To do it, enable these field on the customer portal, but keep them <i>Hidden</i>.</p> <p>For more info, see Configuring Jira Service Management approvals.</p>
Approvers	This a default field and works just like the fields mentioned above. You can keep it, but weve removed it and used the Insight object/s field instead. The Insight field lets you add approvers that are related to the assets youre changing, and they are the best approvers of your requests. Regardless of what youll do with this field, youll also create the Insight field in the next step.

Create new custom fields

The best practice of ITIL is to have the implementation, test, and backout plans for your changes. These fields are text fields and will let you enter and store these plans when creating or editing a change.

Create the custom fields

To create the planning custom fields:

1. Go to **Administration > Custom fields**.
2. Select **Add custom field**.
3. Create the fields as described in the table below.

4. Associate each field with the *Change* issue type and your project.
5. Associate each field with the *Create* and *Edit/View* screens for your *Change* issue type.

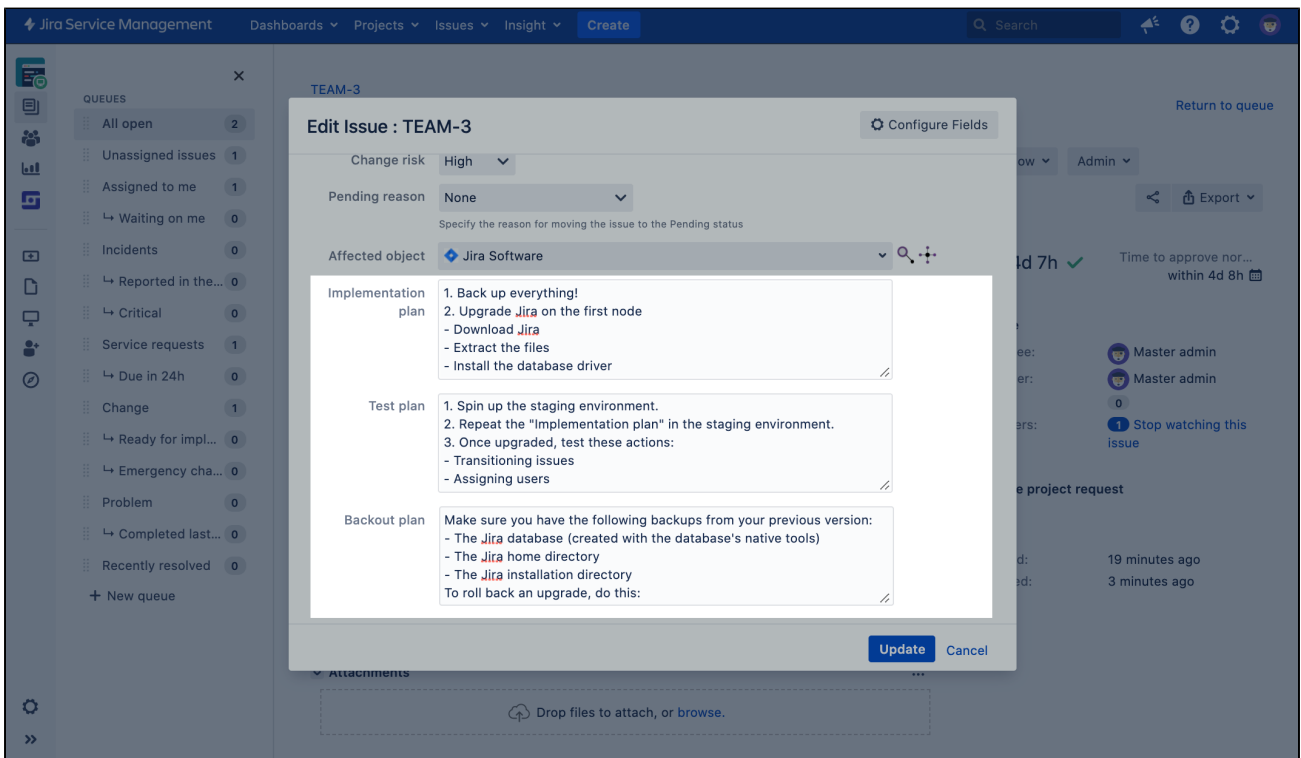
i Just to give you some tips in a default Jira instance, these screens would be called:

- *Project key:* Jira Service Management: Change Create Issue Screen
- *Project key:* Jira Service Management: Change View/Edit Screen

Custom field	Type
Implementation plan	Text field (multi line)
Test plan	Text field (multi line)
Backout plan	Text field (multi line)

View the new fields

Heres how these fields should look on your changes:



Enable the fields on the customer portal

By default, all these custom fields are available only in the agent view. If you'd like customers to create change requests through customer portal, you need to enable and show them on the portal.

To enable fields on the customer portal:

1. Open your project and go to **Project settings > Request types**.
2. Create or select a request that will work as a change request for your customers.
3. Select **Edit fields**.
4. Select **+ Add a field**, and add your fields. After saving, they should appear in the **Visible fields** section.

Next steps

When youre ready, go to [4. Link your project to Insight asset management.](#)

4. Link your project to Insight asset management

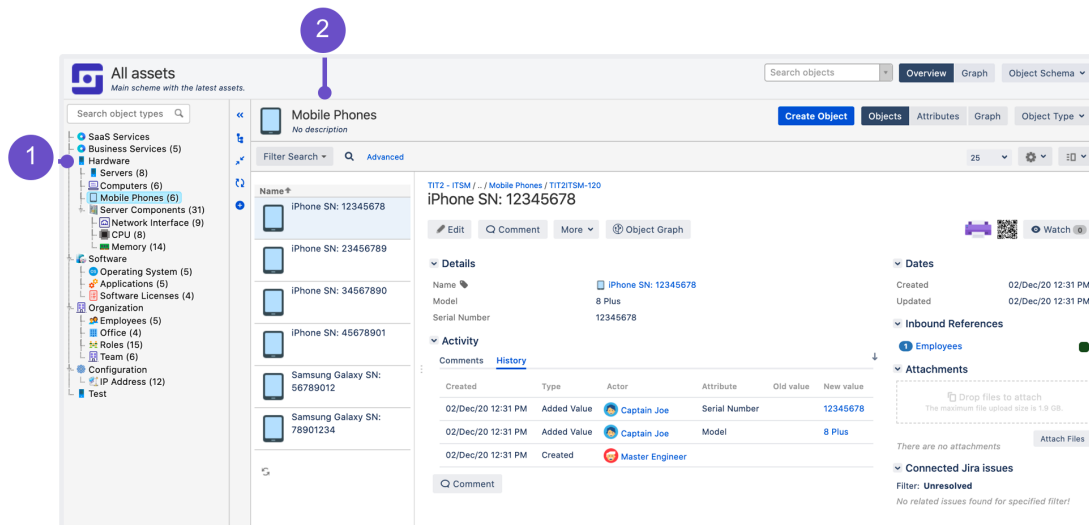
A change request should be opened against something be it a configuration item, physical server, or some other asset that you need to change or improve. Well help you create a basic inventory of assets in Jira Service Management and then link it to your project so you can start including assets in changes.

Over time, you can improve your inventory and make any adjustments to this configuration you need.

Insight at a glance basic overview

i This is a basic overview of Insight, just to show you how it all works and connects together. If you'd like to learn more, [heres a getting started guide](#).

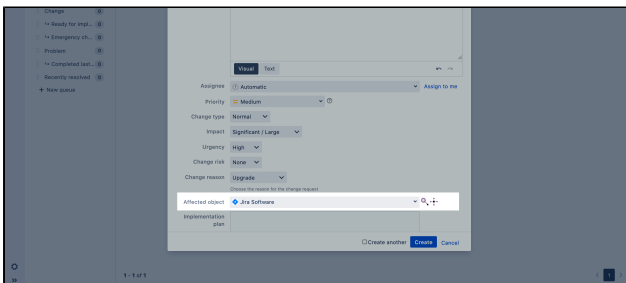
If you're not so familiar with the basics, assets in Insight are called objects. This is what it looks like in Insight:



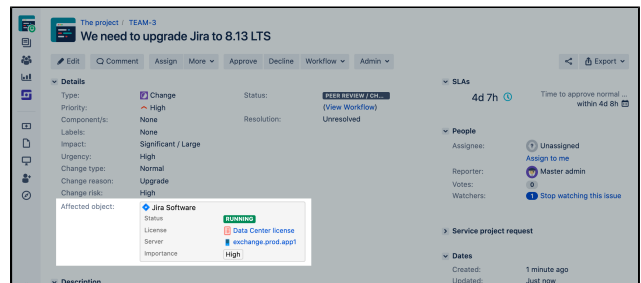
- Object types:** These work like categories that group your actual assets. They would typically be offices, computers, or devices.
- Objects:** These are your assets, grouped within object types. Each asset has attributes that help identify and describe it, just like issue fields.

Linking your inventory and including assets

Objects can be selected and included in issues through Insight custom fields that link projects to your Insight inventory. Such a custom field can be linked to your whole inventory or only parts of it this will affect the objects that can be selected by users.



Selecting an object



Viewing the object in your issue

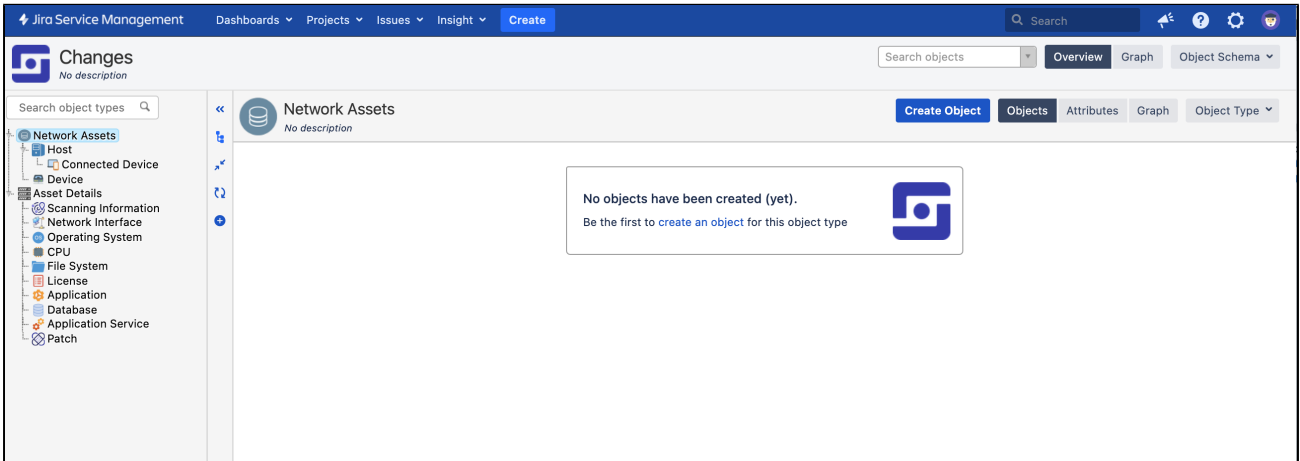
Create a basic Insight inventory

Create the object schema

To create a basic Insight inventory:

1. In the Jira top navigation bar, select **Insight > Insight Object Schemas**.
2. Select **Create Object Schema**.
3. Choose **Create Sample IT Asset Schema**.
4. Give it a name and click **Create**.

Your inventory has been created with a sample structure, but it still needs some objects and adjustments.



Add attributes needed for approvals and automation rules

Approvals and automation rules that you will add in the next steps rely on the *Importance* and *Owner* attributes of your objects in Insight. Lets add these attributes, you can make the adjustments later.

To add attributes:

1. Select the *Applications* object type in the list on the left. You can select any other, but well use this one in the examples.
2. In the top right, select **Attributes**.
3. Add an attribute by specifying fields in the last line of attributes. You can choose a name, type and type value, as well as configure additional items by selecting the cog icon.
4. Click **Add**. Your attribute will be added to the object type, which means it will be used by all objects within this object type.

Need more details? See [Adding attributes to object types](#).

Add the following attributes to the *Applications* object type, which is under *Asset details*. You can choose other object types, but well use this one in examples:

Attribute name	Type and Type value	Values
Importance	<ul style="list-style-type: none"> • Type: Default • Type value: Select 	To add the options, click the cog icon > Configure . <ul style="list-style-type: none"> • Critical • High • Medium • Low
Owner	<ul style="list-style-type: none"> • Type: User or Group 	You can choose the value when you edit an underlying object. Choose a user or group who will be responsible for this particular object.

Create some objects

Add a *Jira Software* object to the *Applications* object type. You can add more objects if you wish.

To create objects:

1. Select any of the object types in the list on the left.
2. In the top right, select **Objects**.
3. Select **Create objects**, and fill in the attributes (fields). All these attributes have been inherited from the parent object type.

Need more details? See [Creating objects](#).

Here's a sample *Jira Software* object you can create:

Object type: Asset details > Applications.

Object details:

- Name: Jira Software
- Vendor: Atlassian
- Version: LTS
- Importance: High
- Owner: jira-administrators, or any other user

Link your inventory to your Jira project

Your Insight inventory can be linked to Jira projects by using the Insight custom fields. They behave like regular Jira custom fields but have additional configuration that lets you map it to your inventory, or some parts of it.

 For more details, see [Adding Insight custom fields to screens in Jira](#).

Create the Insight object/s custom field

To create the Insight custom field:

1. Go to **Administration > Custom fields**.
2. Select **Add custom field**.
3. Select **All**, and choose **Insight Object/s**.
4. Give it a name *Affected object* and associate the field with change requests and your project.
5. Click **Create**.
6. Associate the field with the *Create* and *View/Edit* screens, like you did in one of the previous steps.

Configure the field

We'll map the field to your whole inventory, just to have access to all assets. You can adjust it later if you wish to.

To configure the Insight custom field:

1. On the Custom fields page, search for your custom field.
2. Select **> Configure**.
3. Select **Edit Insight Configuration**.
4. Set **Object Schema** to the object schema you created. That's your inventory.
5. Leave the remaining fields as they are.

Configure Insight Custom Field - Affected object (customfield_10134)

Scope

Object Schema	All assets	ⓘ
Filter Scope (IQL)	Not configured	ⓘ
Filter Issue Scope (IQL)	Not configured	ⓘ
Filter Assign Scope (IQL)	Not configured	ⓘ
Make Filter Assign Strict	<input type="checkbox"/>	ⓘ

User Interaction

Type of interaction	<input checked="" type="radio"/> Select <input type="radio"/> Checkbox / Radio button	ⓘ
Filter objects with attributes	Label x	ⓘ
Object attributes on Issue view		ⓘ
Multiple	<input type="checkbox"/>	ⓘ
Option to add all objects	<input type="checkbox"/>	ⓘ

Jira Service Management

Show object picker on customer portals	<input checked="" type="checkbox"/>	ⓘ
Force to show on Customer Issue details	<input type="checkbox"/>	ⓘ

[Return to Jira Custom Field](#)

Enable the field on the portal

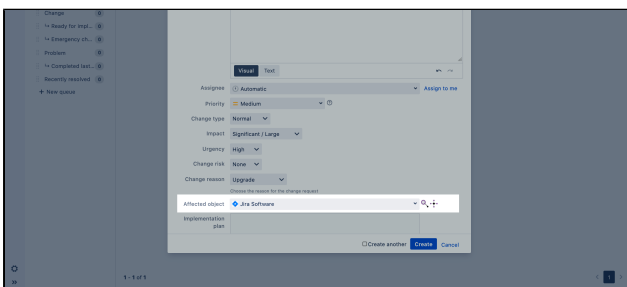
Like with other custom fields, if you'd like your customers to be able to select a value for this field, you need to enable and show it on the customer portal.

To enable the field on the customer portal:

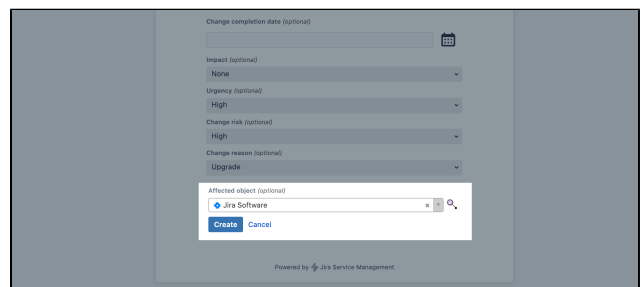
1. Open your project and go to **Project settings > Request types**.
2. Find a request that works as a change request for your customers.
3. Select **Edit fields**.
4. Select **+ Add a field**, and add your field. After saving, it should appear in the **Visible fields** section.

View the field

When you create a new change, your custom field should be displayed on the screen. You should also be able to select objects from your inventory.



Agent view



Customer portal

Next steps

When you're ready, go to [5. Add approvals to your workflow](#).

5. Add approvals to your workflow

Every change should be approved and accounted for. In this step, we'll show you how to add approvers to your workflows. We'll use approvers from Insight objects (people responsible for your assets) and Jira users (CAB and change managers), and we'll add them to two workflow steps.

Add approvals to your workflow

To add an approval:

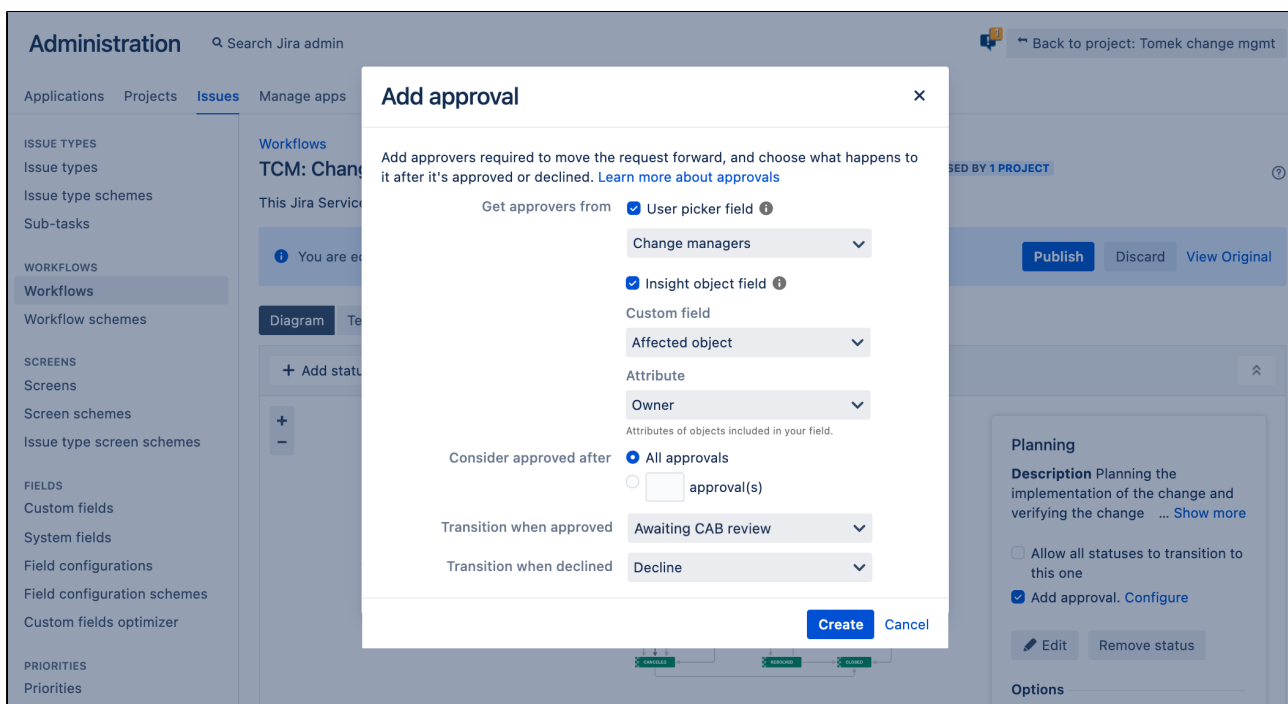
1. In your project, go to **Project settings > Workflows**.
2. Select your change workflow and edit it.
3. Select the right workflow steps, as described in the next sections. You'll do it separately for the **Approving** and **CAB review** steps.
4. Select **Add approval**, and click **Configure** next to it.



Workflow step: Approving

This workflow step is only used by *Normal* changes. It's an initial review by change managers and asset owners that are overseeing changes made to production services.

Configure approval for the **Approving** step in the following way:

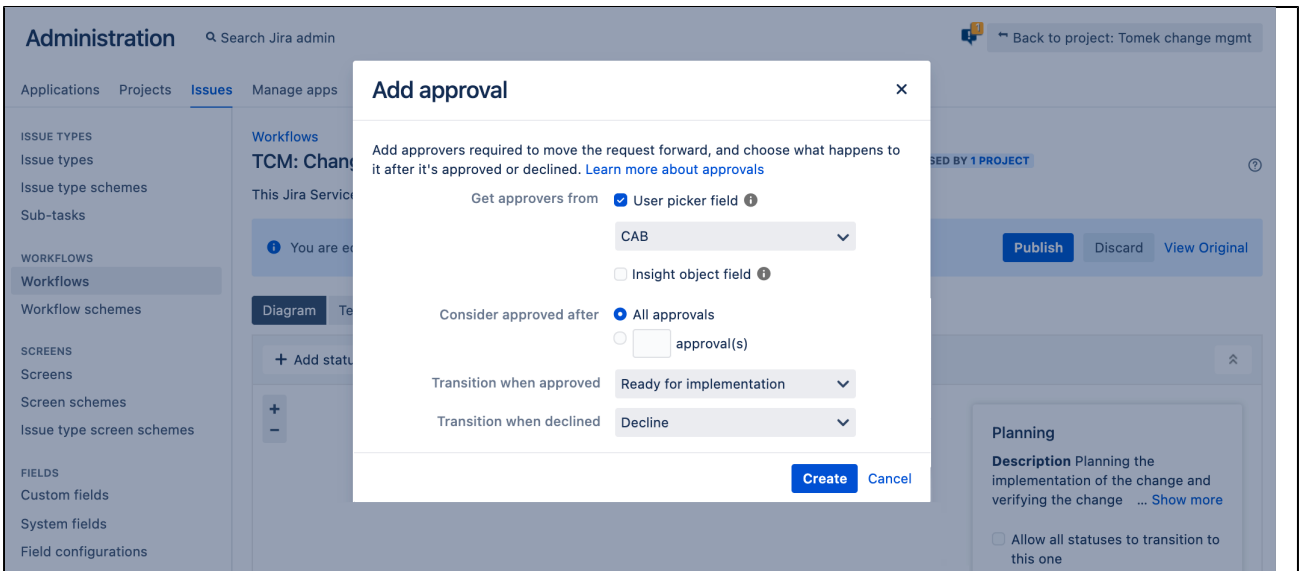


Field	Value
Get approvers from	User picker field <ul style="list-style-type: none"> • Custom field: Change managers
	Insight object field <ul style="list-style-type: none"> • Custom field: Affected object • Attribute: Owner
Consider approved after	Up to you, usually 1 or 2 approvals are enough.
Transition when approved	Awaiting CAB review Note: Not all teams are using CAB reviews and we also treat it as an optional step. If you'd like to omit it, you can set this transition to <i>Ready for implementation</i> .
Transition when declined	Decline

Workflow step: CAB review

This workflow step is only used by *Normal* changes. It's a thorough review by Change Advisory Board (CAB) to assess the change and reduce the potential risk.

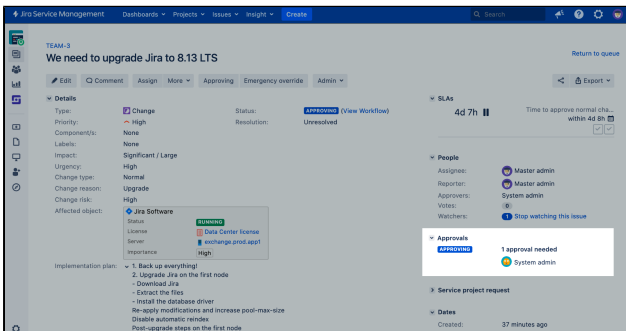
Configure approval for the **CAB review** step in the following way:



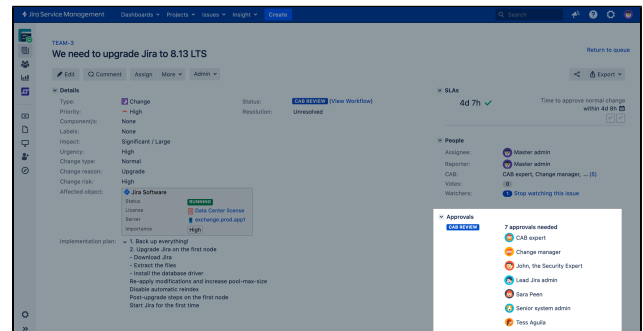
Field	Value
Get approvers from	User picker field <ul style="list-style-type: none"> • Custom field: CAB
Consider approved after	Up to you, usually 1 or 2 approvals are enough.
Transition when approved	Ready for implementation
Transition when declined	Decline

Result

Here's a sample change request with approvals, in two different workflow steps that you configured:



Approving



CAB review

Next steps

When you're ready, go to 6. Create and update automation rules.

6. Create and update automation rules

Automation rules will do the repetitive tasks for you and make sure your change requests behave as they should. The rules you'll create will do the following actions:

- Determine and assign the proper change risk
- Calculate the priority
- Escalate Emergency changes
- Auto-approve Standard changes

Create automation rules

1. Open your project.
2. Go to **Project settings > Automation**.
3. Select **Add rule** and choose **Custom rule**.
4. Add the following rules:

Rule 1: Determine risk based on affected objects importance

This rule will check the value of the *Importance* attribute of the affected Insight object and adjust the change risk accordingly. For example, if the objects importance is Critical, the risk will be set to Critical, and if its High the risk will be set to High as well.

When...	If...	Then...
Trigger: Issue created	Condition: Issue matches Advanced query: <pre>issuetype = Change AND "Affected object" in iqlFunction ("Importance = \"Critical\"")</pre>	Action: Edit issue Field: Change risk Replace with value: Critical
	Else if...	Then...
	Condition: Issue matches Advanced query: <pre>issuetype = Change AND "Affected object" in iqlFunction ("Importance = \"High\"")</pre>	Action: Edit issue Field: Change risk Replace with value: High
	Else if...	Then...

	<p>Condition: Issue matches</p> <p>Advanced query:</p> <pre>issuetype = Change AND "Affected object" in iqlFunction ("Importance = \"Medium\"")</pre>	<p>Action: Edit issue</p> <p>Field: Change risk</p> <p>Replace with value: Medium</p>
	Else if...	Then...
	<p>Condition: Issue matches</p> <p>Advanced query:</p> <pre>issuetype = Change AND "Affected object" in iqlFunction ("Importance = \"Low\"")</pre>	<p>Action: Edit issue</p> <p>Field: Change risk</p> <p>Replace with value: Low</p>

Rule 2: Calculate priority based on changes impact and urgency

This rule will check the values of the *Impact* and *Urgency* fields of the change and update the priority accordingly.

When...	If...	Then...
<p>Trigger: Issue created</p>	<p>Condition: Issue matches</p> <p>Advanced query:</p> <pre>issuetype = Change AND (Impact = "Extensive / Widespread" AND Urgency in (High, Critical)) OR (Impact = "Significant / Large" AND Urgency = Critical)</pre>	<p>Action: Edit issue</p> <p>Field: Priority</p> <p>Replace with value: Critical</p>
	Else if...	Then...
	<p>Condition: Issue matches</p> <p>Advanced query:</p> <pre>issuetype = Change AND (Impact = "Moderate / Limited" AND Urgency = Critical) OR (Impact = "Significant / Large" AND Urgency = High) OR (Impact = "Extensive / Widespread" AND Urgency = Medium)</pre>	<p>Action: Edit issue</p> <p>Field: Priority</p> <p>Replace with value: High</p>
	Else if...	Then...

<p>Condition: Issue matches</p> <p>Advanced query:</p> <pre>issuetype = Change AND (Impact = "Minor / Localized" AND Urgency = Critical) OR (Impact = "Moderate / Limited" AND Urgency in (High, Medium)) OR (Impact = "Significant / Large" AND Urgency = Medium) OR (Impact = "Extensive / Widespread" AND Urgency = Low)</pre>	<p>Action: Edit issue</p> <p>Field: Priority</p> <p>Replace with value: Medium</p>
<p>Else if...</p>	<p>Then...</p>
<p>Condition: Issue matches</p> <p>Advanced query:</p> <pre>issuetype = Change AND (Impact = "Minor / Localized" AND Urgency in (High, Medium, Low)) OR (Impact in ("Moderate / Limited", "Significant / Large") AND Urgency = Low)</pre>	<p>Action: Edit issue</p> <p>Field: Priority</p> <p>Replace with value: Low</p>

Rule 3: Escalate emergency changes

This rule will transition Emergency changes right to the *Implementation* status on the workflow. Also, a public comment about it will be added.

When...	If...	Then...
<p>Trigger: Issue created</p>	<p>Condition: Issue matches</p> <p>Advanced query:</p> <pre>issuetype = Change AND "Change type" = Emergency</pre>	<p>Action: Transition issue</p> <p>Transition: Emergency override</p> <hr/> <p>Action: Add comment</p> <p>Comment text:</p> <pre>This is an automated message to tell you that your request has been transitioned to Implementing.</pre> <p>Comment type: Public</p>

Update existing automation rules

Edit the following rule to make it work with the new workflow.

Rule: Auto-approve standard changes

With these updates, Standard changes will be automatically approved and transitioned to the Implementing status. Also, a public comment about it will be added.

When...	If...	Then...
Trigger: Issue created	Condition: Issue matches	Action: Transition issue Transition: Ready for implementation
	Advanced query: <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> issuetype = Change AND "Change type" = Standard </div>	Action: Auto approve/decline Answer: Approve
		Action: Add comment Comment text:
		<div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> This is an automated message to tell you that your request has been pre-approved. </div> Comment type: Public

Next steps

When you're ready, go to [7. \(Optional\) Set up a calendar to coordinate your changes.](#)

7. (Optional) Set up a calendar to coordinate your changes

Team Calendars is part of Confluence Data Center, and you can optionally use it to view your planned changes in a calendar. It will let you track changes and schedule them so they coincide with business events. [Learn more about Team Calendars](#)

To set up a change calendar with Confluence:

1. In Confluence, go to your teams space.
2. From the sidebar, select **Calendars**.
3. Select **Add calendar**.
4. Viewing the calendar, select **Add event**.
5. Select the **Event Type** drop down and choose **Jira Issue Dates**.
6. Under **Display**, select the **JQL (advanced)** and enter the following:

```
project = "Your IT service desk project name" AND issuetype= Change
```

7. Under **Date range**, select **Add start and end date**. Select **Change start dates** as the start date. Select **Change completion dates** as the end date.
8. Select **OK**.

The calendar automatically picks up the start and end dates of change requests from your service project. Then, it plots them on the calendar.

The screenshot shows the 'My Calendars' interface. At the top, there's a title 'My Calendars' and a '+ Add Calendar' button. Below that, there are navigation controls for 'Today', 'April 2016', and view options: 'Month', 'Week', 'List', and 'Timeline'. On the right, there are buttons for 'Add Event' and 'Change schedules'. The calendar grid shows dates from Monday, April 28th to Sunday, April 3rd. Two events are plotted: a green bar for '11:00 PM CMONE-4 - LDAP DB Update' on Friday, April 8th, and another green bar for '5:30 PM CMONE-3 - Hardware upgrade for rack 1' on Saturday, April 9th. A tooltip for the Saturday event provides details: 'CMONE-3 - Hardware upgrade for rack 1', 'Rack 1 requires new hardware upgrade to increase performance', 'Change Start: 4 days remaining', 'Assignee: Unassigned', and 'Status: Peer Review'. There are also links for 'Edit' and 'View Issue'.

You're good to go

These are all the steps needed to adjust your project to new change management. If you haven't checked it already, see [Sample lifecycle of a change](#) for an overview of how your changes will behave.

Managing incidents with your IT service desk

An incident model helps service desks investigate, record, and resolve service interruptions or outages. An Information Technology Infrastructure Library (ITIL) incident management workflow aims to reduce downtime and negative impacts.

Incident management focuses on short-term solutions. To manage occurring incidents or underlying problems, see [Managing problems with your IT service desk](#).

The IT Service Desk template comes with an incident management workflow. This workflow ensures that you log, diagnose, and resolve incidents. We recommend you start with this workflow and adapt it to your business needs.

When managed well, incident records can identify:

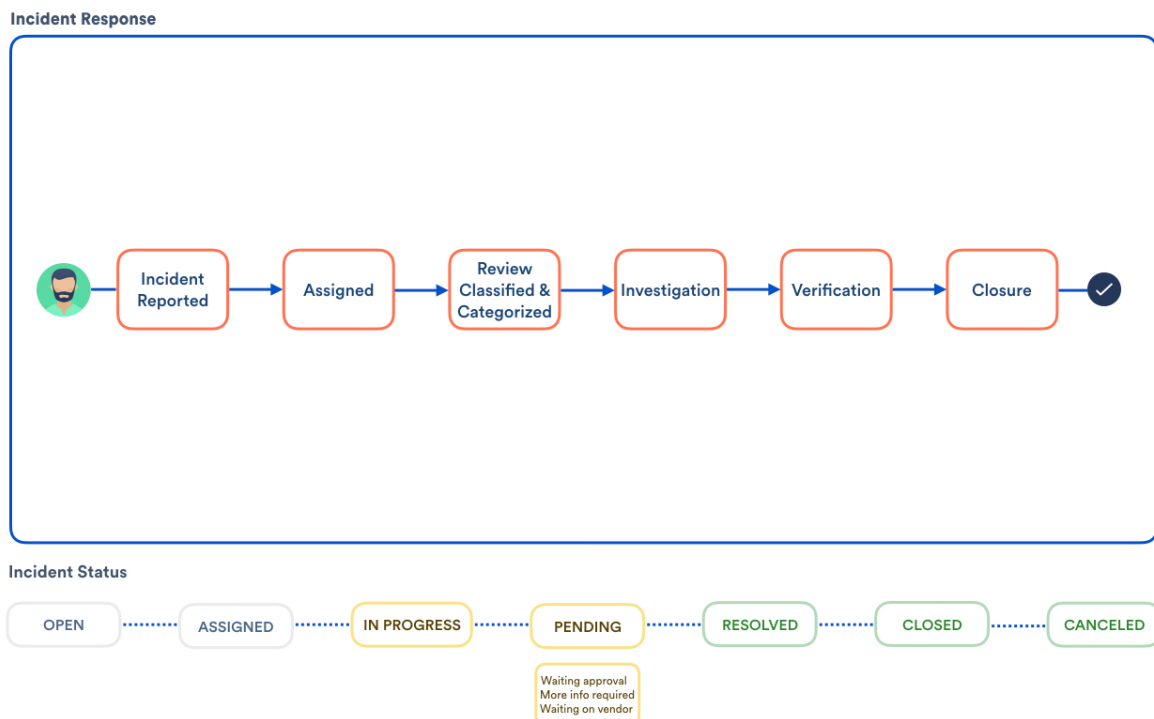
- missing service requirements
- potential improvements
- future team member training

This page describes some best practices for managing incidents using Jira Service Management. You may seek formal training in how to make ITIL recommendation work best for your business.

Incident management process

The IT Service Desk template associates certain requests with an incident workflow. We set up the workflow to complement the following incident management process. We recommend you start with this workflow and adapt it to your specific needs over time.

The ITIL incident management process, in brief:



1. Service end users, monitoring systems, or internal IT members report interruptions.
2. The service desk describes and logs the incident. They link together all reports related to the service interruption.
3. The service desk records the date and time, reporter name, and a unique ID for the incident. Jira Service Management does this automatically.
4. A service project agent labels the incidents with appropriate categorization. The team uses these categories during post-incident reviews and for reporting.
5. A service desk agent prioritizes the incident based on impact and urgency.

6. The team diagnoses the incident, the services effected, and possible solutions. Agents communicate with incident reporters to help complete this diagnosis.
7. If needed, the service project team escalates the incident to second-line support representatives. These are the people who work regularly on the effected systems.
8. The service project resolves the service interruption and verifies that the fix is successful. The resolution is fully documented for future reference.
9. The service desk closes the incident.
10. Team members should carry out post-incident reviews for major incidents. These investigations can help determine:
 - a. missing requirements
 - b. potential changes to service level agreements
 - c. potential service improvements or focus areas

Set up incident management in Jira Service Management

Configure the workflow and fields with the Incident Management workflow add-on

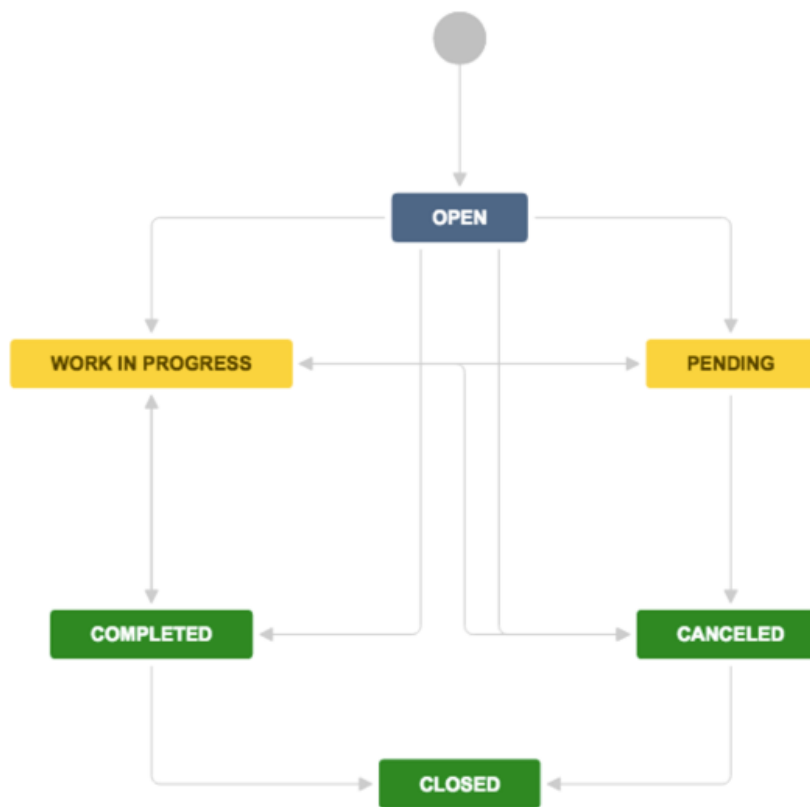
We used the ITIL framework to build the following workflow add-on for incident management: <https://marketplace.atlassian.com/plugins/com.atlassian.servicedesk.incident/server/overview>.

You can use this workflow as a template for your own incident management process.

To use the workflow from the Marketplace:

1. Log in as a user that has the Jira administrator global permission, and follow the instructions listed here to [import a workflow](#).
2. To add the workflow fields to your incidents, activate the screen by following the instructions here: [Defining a screen](#).

Incident management workflow



Default form fields for incident reports

Jira Service Management allows you to customize the fields of information collected from customers. Additionally, you can customize the fields of information used by your agents. Jira Service Management does this through issue type fields and screens. Fields help agents investigate, assess, and categorize the incident for reporting or querying.

By default, we include the following fields in your agents' view of an incident. If needed, you can add custom fields. [Find out more about fields in Jira.](#)

Field name	Description
Summary	A short description of the request.
Reporter	The person who submitted the request.
Component/s	Segments of your IT infrastructure that relate to the request. For example, "Billing services" or "VPN server". These are used for labeling, categorization, and reporting.
Attachment	Files or images added to the request.
Description	A long, detailed description of the request.
Linked Issues	A list of other requests that affect or are effected by the request. If your business uses other Atlassian products, this list may include linked development issues.
Assignee	The team member assigned to work on the request.
Priority	The importance of the request's resolution, usually in regards to your business needs and goals. Sometimes, priority is calculated by impact and urgency.
Labels	A list of additional custom labels used for categorizing or querying records.
Request participants	A list of extra customers who take part in the request, for example, people from other teams or vendors. Read more about participants.
Approvers	A list of people responsible for approving the request, usually business, financial or technical contacts .
Organizations	A list of customer groups interested in the request's resolution.
Impact	The effect of the incident, usually in regards to service level agreements.
Urgency	The time available before the business feels the incident's impact.
Pending reason	A short description or code that indicates why the incident is not progressing.
Product categorization	A category of IT asset or system that the request effects.
Operational categorization	A category of action or function required to fulfill the request.
Source	The asset or system where the incident originated.

Learn more about IT service management (ITSM)

Get more tips and tricks for successful [ITSM](#), view case studies, and learn how to take your service project to the next level.

[Check out the ITSM resources on IT Unplugged.](#)

Managing problems with your IT service desk

Information Technology Infrastructure Library (ITIL) distinguishes between incidents and problems. Incident management serves to quickly restore services or broken experiences. For more information on incidents, see [incident management](#).

Problem management seeks to prevent incidents from happening again. Problems are typically reported by internal IT members and not their customers.

An ITIL problem management workflow aims to investigate, record, and prevent IT infrastructure problems. The IT Service Desk template comes with a built-in workflow for handling problems. We recommend you start with the template's default workflow and adapt it to your business needs. When correctly managed, problem records prompt agents to detail known errors and workarounds in your knowledge base. These documents:

- help service agents resolve issues and restore services
- reduce downtime
- increase the quality and trust of your IT infrastructure

This page describes best practices for managing problems using Jira Service Management. You may seek formal training in how to make ITIL recommendation work best for your business.

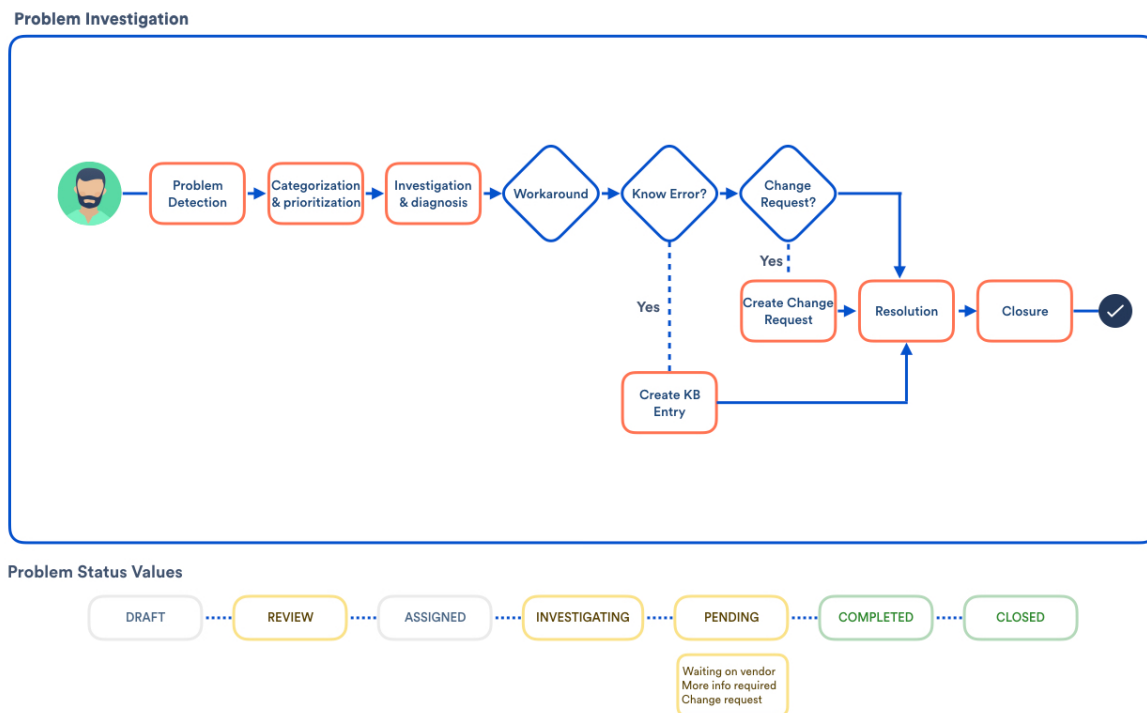
Problem management process

The IT Service Desk template comes with a problem-management workflow. We set up this workflow to complement the following problem management process. Use the workflow to transition problem registers alongside these ITIL recommended activities:

- problem investigation
- identification of workarounds
- recording of known errors

We recommend you start with our default workflow and adapt it to your specific needs over time.

The ITIL problem management process, in brief:



1. Incident trends, vendors, or technical support staff report problems to the service desk.
2. A service desk team member records the details of the problem and links all related incidents.

3. A service desk agent labels the problem with appropriate categorization. They may reuse the labels of the incidents linked to the problem. The team uses these categories during review and for reporting.
4. A service desk agent prioritizes the problem. They base priority on the frequency of related incidents and their impact.
5. The service desk team determines the root cause of the problem.
6. The service desk team records the workarounds used to resolve related incidents. These workarounds to reduce service interruptions until the service desk fully resolves the problem.
7. The service desk team adds known errors to their knowledge base. They include symptoms of related incidents and relevant workarounds.
8. The service desk team proposes a change to the infrastructure to resolve the problem.
9. The service desk closes the problem.
10. Team members should carry out in-depth reviews of major problems.

Set up problem management in Jira Service Management

Configure the workflow and fields with the problem management workflow add-on

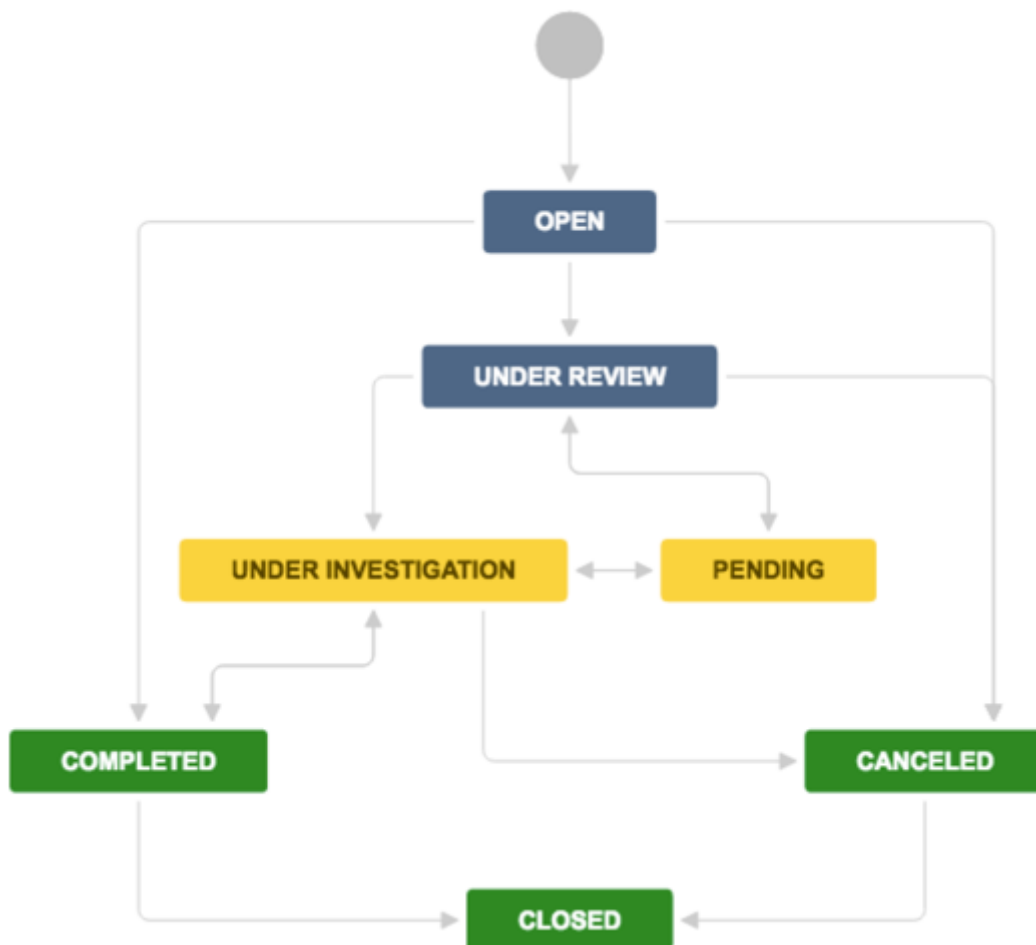
We used the ITIL framework to build the following workflow add-on for problem management: <https://marketplace.atlassian.com/plugins/com.atlassian.servicedesk.problem> .

You can use this workflow as a template for your own problem management process.

To use the workflow from the Marketplace:

1. Log in as a user that has the Jira administrator global permission, and follow the instructions listed here to [import a workflow](#).
2. To add the workflow fields to your incidents, activate the screen by following the instructions here: [Defining a screen](#).

Problem management workflow



Default form fields for problem reports

Jira Service Management allows you to customize the fields of information collected from customers. Additionally, you can customize the fields of information used by your agents. Jira Service Management does this through issue type fields and screens. Fields help agents investigate, assess, and categorize the problems for reporting or querying.

By default, we include the following fields in your agents' view of a problem. If needed, you can add custom fields. [Find out more about fields in Jira.](#)

Field name	Description
Summary	A short description of the request.
Reporter	The person who submitted the request.
Component /s	Segments of your IT infrastructure that relate to the request. For example, "Billing services" or "VPN server". These are used for labeling, categorization, and reporting.
Attachment	Files or images added to the request.
Description	A long, detailed description of the request.
Linked Issues	A list of other requests that affect or are effected by the request. If your business uses other Atlassian products, this list may include linked development issues.
Assignee	The team member assigned to work on the request.
Priority	The importance of the request's resolution, usually in regards to your business needs and goals. Sometimes, priority is calculated by impact and urgency.
Labels	A list of additional custom labels used for categorizing or querying records.
Request participants	A list of extra customers who take part in the request, for example, people from other teams, or vendors. Read more about participants.
Approvers	A list of people responsible for approving the request, usually business, financial or technical contacts .
Organizations	A list of customer groups interested in the request's resolution.
Impact	The effect of the problem, usually in regards to service level agreements.
Urgency	The time available before the business feels the problem's impact.
Source	The asset or system where the problem originated.
Investigation reason	The trigger for prompting an investigation. For example, reoccurring incidents, non-routine incidents, or other.
Pending reason	A short description or code that indicates why the problem is not progressing.
Product categorization	A category of IT asset or system that the request effects.
Operational categorization	A category of action or function required to fulfill the request.
Root cause	The original cause of the incidents related to the problem.

Workaround	The detailed description of a temporary, known solution to restore a service. If you have Confluence, we recommend you document your workaround in your knowledge base.
------------	---

Learn more about IT service management (ITSM)

Get more tips and tricks for successful [ITSM](#), view case studies, and learn how to take your service desk to the next level.

[Check out the ITSM resources on IT Unplugged.](#)

Calculating priority automatically

Jira Service Management comes with some powerful automation tools. IT teams can set up their service project to calculate a request's priority automatically.

Removing manual processes gives time back to your team. Your team spends less time triaging and prioritizing requests. And, they spend more time resolving IT service tasks. Calculating the correct priority helps put requests into the correct SLA.

Some IT teams use an impact-urgency matrix to determine the priority of an issue. This page walks through an example for defining this matrix. Then, it discusses using these decisions to automate how priority calculation with these fields.

At the end, you should know a bit more about automation and how it can help you remove all sorts of manual processes. Automation frees up your agents' time and makes your service desk more efficient.

Create a priority matrix using impact and urgency values

Work with your team to determine how your service desk prioritizes incidents. Below is a sample matrix for how our team thinks about priority. Yours may differ depending on your resources and other factors.

Here's an example matrix:

Impact	Urgency			
	Critical	High	Medium	Low
Extensive / Widespread	Highest priority	Highest priority	High priority	Medium priority
Significant / Large	Highest priority	High priority	Medium priority	Low priority
Moderate / Limited	High priority	Medium priority	Low priority	Lowest priority
Minor / Localized	Medium priority	Low priority	Lowest priority	Lowest priority

Note that the priority values listed above are just examples for this tutorial. You can also create priorities that are specific to your service desk. See [Defining priority field values](#) and [Associating priorities with projects](#) for more details.

Before you begin

Make sure you have the following two [custom fields](#) of the type **Select List (single choice)**, each containing the values listed in the previous table:

- **Urgency:** Critical, High, Medium, Low
- **Impact:** Extensive, Significant, Moderate, Minor

Automating the calculation

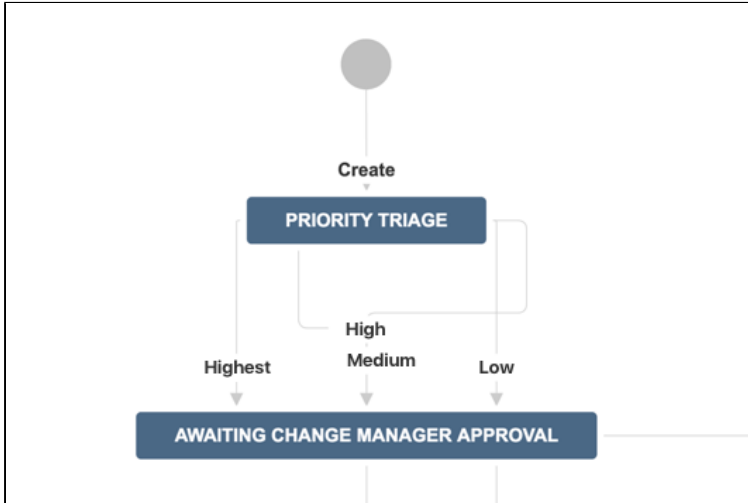
After Step 1, the workflow transitions a change to **Priority triage** status as soon as an issue is created. You will also have 4 transitions to set the **Priority** to a value.

With Step 2, you will set up an automation rule that checks the value of the Urgency and Impact fields and fires off the corresponding transition according to the matrix.

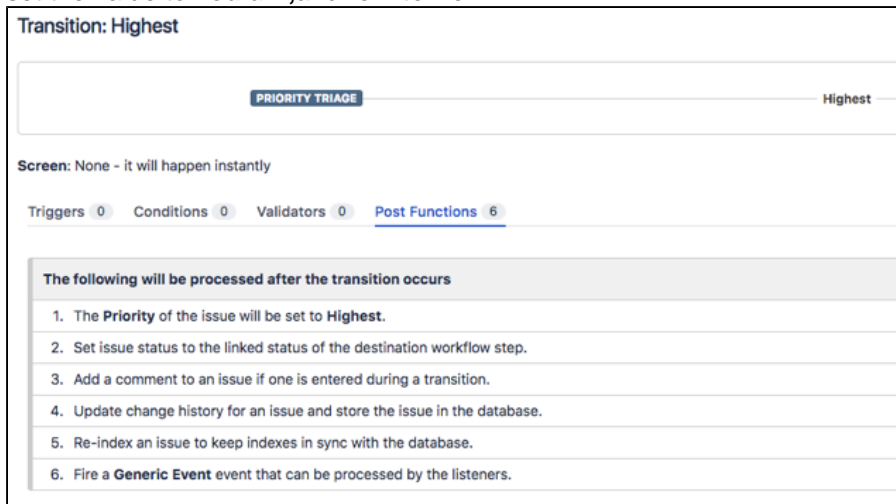
Step 1: Configure the workflow

1. Go to the workflow that is used by your **Change** issue type.
2. Between the **Create** transition and your first status, add a new status and name it **Priority Triage**.
3. Add the following four transitions from this status:
 - a. Highest
 - b. High

- c. Medium
- d. Low



4. In each of these transitions, add a post function.
 - a. Select the **Update Issue Field** post function.
 - b. In the post function, update the **Priority** field to match the transition, for example, the *Highest* transition will have a post function that changes the **Priority** field to **Highest**. Similarly, the **High** transition will have this post function to set the **Priority** field as **High**, and the **Medium** transition to set the value to **Medium**, and **Low** to **Low**.



5. Publish the workflow.

Step 2: Configure the automation rule

After the post function is set, let's create an automation rule that triggers the appropriate transition depending on the urgency and impact selected during request creation.

The rules should follow the following pattern:

- **When:** Issue created
- **If, or Else if:** Specify the urgency and impact value pair according to the matrix, for example:

```
status = "Priority triage" AND Impact = "Extensive / Widespread" AND Urgency = Highest
```

- **Then:** Transition issue, and select the transition that matches the value pair according to the matrix.

Note: If your **Urgency** or **Impact** value is optional on the request type form or issue create screen, then there might be cases where these fields are empty. In this case, make sure that you add a **Else if** condition that caters for this scenario, for example:

- **Else if:**


```
status = "Priority triage" AND Impact is empty OR Urgency is EMPTY
```

- **Then:** Transition issue, **Low**.

Learn more about IT service management (ITSM)

Get more tips and tricks for successful [ITSM](#), view case studies, and learn how to take your service desk to the next level.

[Check out the ITSM resources on IT Unplugged.](#)

Manage changes with your IT service management project 1

Effective service teams plan and control changes, as they understand their impact to their business. Although Jira Service Management provides default change management workflows and fields, weve now updated them to better follow the Information Technology Infrastructure Library (ITIL) best practices. Weve also added a few additional steps that will let you create an inventory of assets and open changes against them by using the built-in asset management functionality.

i Learn more about ITIL and change management

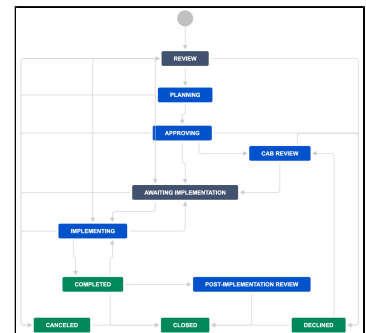
If youre not familiar with the ITIL best practices, check out our resources. Theyll explain you the basic concepts, definitions, and processes. [Learn more about change management](#)

Set up change management in Jira Service Management

Have a look at the list of customizations weve made to the change management process. If youd like to use them, youll need to adjust your individual projects manually.

Change management workflow

The workflow follows the ITIL best practices and terminology, and makes sure your changes are approved and accounted for. Extra steps let you properly plan and assess the most important and risky changes, at the same time not slowing down critical changes that must be implemented right away. All this with the right stakeholders sharing their expertise and approving your changes along the way.

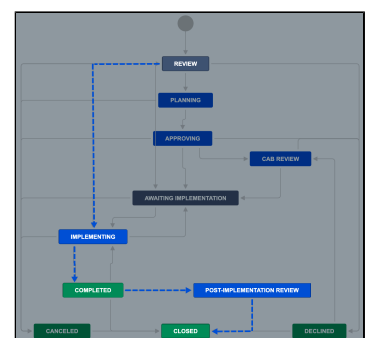


Different types of changes using different workflow paths

Thanks to automation rules, different types of changes can take different paths. Just so emergency changes dont have to wait for all of the approvals theyre emergencies after all.

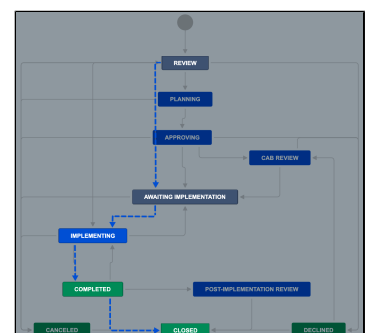
EMERGENCY

An emergency change must be assessed and implemented as quickly as possible. Such a change omits a few steps and goes right into implementation, but it includes an extra review once it's done.



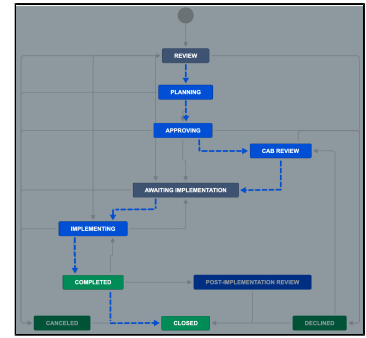
STANDARD

A standard change occurs frequently, is low risk, and has a pre-established procedure with documented tasks for completion. As such, it needs less planning and fewer approvals.



NORMAL

A normal change typically requires an important change to a service or infrastructure, but you might not have a well established process for it. Its good to properly plan and approve it.



Automation rules handling repetitive tasks for you

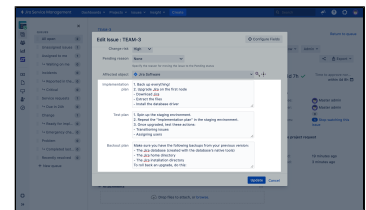
Automation rules will also take some burden off your back and automate repetitive tasks, just so you dont have to think about them:

- Determine change risk based on the importance of affected assets
- Calculate priority based on Impact and Urgency
- Escalate Emergency changes
- Auto-approve Standard changes

Change plans

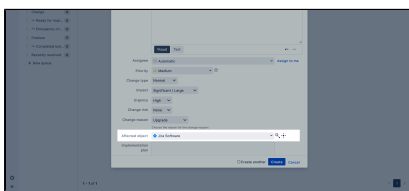
With custom fields, each change requests can store detailed plans, just to make sure everyone working on it knows what theyre doing or can quickly roll back if it turns out they dont.

- Implementation plan
- Test plan
- Backout plan

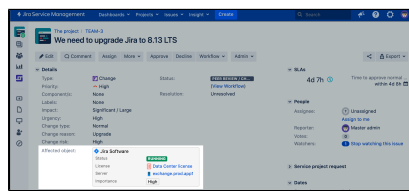


Inventory of assets and configuration items

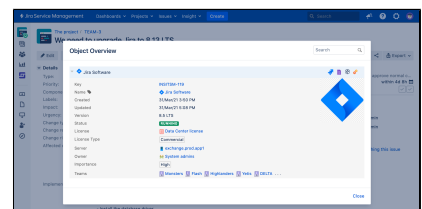
Jira Service Management lets you create an inventory of assets and configuration items, together with their details and relations between them. You can then open changes against these assets and get them linked together. Thanks to that, details about an asset are included in the change request and everyone involved has enough context to decide how risky or important a change is.



Selecting an asset



Viewing an asset

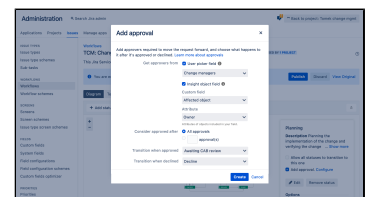


Viewing more details

Approvals from the right stakeholders

Make sure the most important and risky changes are approved by the right people. You can include change managers, CAB, and approvers that are directly responsible for your assets the right people at the right time. By using Insight custom fields, these approvers will change dynamically depending on the asset affected by your change request.

- Change managers (user picker field)
- Change Advisory Board (user picker field)
- Owners of your Insight assets (Insight field)



Get started with improved change management

To get started, follow these steps. They'll help you adjust your projects to all of these changes and provide some explanations about what these changes do:

Best practices for software teams using Jira Service Management

 This guide helps teams using server versions of Jira Service Management and Jira Software.

Software teams that develop with Jira Software can give technical support and get customer feedback using Jira Service Management. Teams using Jira Software can view and comment on Jira Service Management bug reports *without any additional licensing or billing cost*. Read more about [collaborating on Jira Service Management](#).

You can use this guide and Jira Service Management's customer service project template to:

- collect user bugs and feedback via a built-in web portal
- give users technical support
- help users with licensing and billing

Jira Service Management is highly configurable, and the template is just a starting point. Over time, you can fine-tune the template to your specific business needs.

This guide covers how to:

- [Get set up for customer service](#)
- [Collect effective bug reports from customers](#)
- [Customize Jira Service Management's bug report workflow](#)
- [Collaborate with other Jira teams on Jira Service Management issues](#)
- [Escalate Jira Service Management issues to other Jira teams](#)

Collaborate with other Jira teams on Jira Service Management issues

Software teams can get context from your service project agent's conversations with customers. You can set up your service project to allow Jira Software teams to view and comment on Jira Service Management issues.

By default, Jira Service Management agents can:

- view Jira Software issues
- comment on Jira Software issues
- transition Jira Software issues

By default, Jira Software and Jira Core users:

- *can't* view Jira Service Management requests in the customer portal
- *can't* view Jira Service Management issues in a Jira Service Management project
- *can't* comment on Jira Service Management requests in the customer portal
- *can't* comment on Jira Service Management issues in a Jira Service Management project
- *can't* view links to Jira Service Management issues in Jira Software or Jira Core project

If you plan to work with other Jira teams, you should change your project's permissions. We recommend giving all logged-in users permission to view and comment on service desk projects.

The table below contains a breakdown of three different types of permissions. These permissions make collaboration between service desk agents and software development teams possible:

	Request participants	View permissions on Jira Service Management projects	Comment permission on Jira Service Management projects
Permissions	These users can: <ul style="list-style-type: none"> • view requests in the portal • add public comments in the portal • add attachments in the portal • transition the request through customer visible statuses in the portal 	These users can: <ul style="list-style-type: none"> • view issues in service projects • view linked service project issue information in Jira Software projects 	These users can: <ul style="list-style-type: none"> • add internal comments on service project issues
Effect on Jira Service Management licensing and billing	None	None	None

The easiest way to collaborate (out of the box)

Jira Service Management agents can add other Jira team members as request participants. Then, Jira Software or Jira Core members can interact with customers in the web portal.

Request participants can:


- view requests in the portal
- add public comments in the portal
- add attachments in the portal
- transition the request through customer visible statuses in the portal

Agents can add request participants to service desk issues. Look for the **People** section in the issue and select the **Request participants** field to add members from other Jira teams.

[Read more about request participants.](#)

Modify your permissions to let other teams view and comment on service project issues

To view and edit the permission scheme for your service project, go to **Project settings** () > **Permissions**.

 You need to be a Jira admin to change project permission schemes. Changes to permission schemes affect all projects that share that scheme. Be careful!

[Read more about configuring project permission schemes.](#)

Give developers permission to view service project issue

To let all Jira Server users on your site view issues on your service project:

1. On the project permission scheme page, select **Actions > Edit permissions**.
2. Under **Project Permissions**, select **Edit** in the **Browse Projects** entry.
3. In the **Granted** to selection, choose **Application access**.
4. From the dropdown, select **Any logged in user**.
5. Select **Grant**.

This allows other Jira users to:

- view service project issues
- view links to service project requests in their own projects

Give developers permission to comment on service project issues

To let all Jira Server users on your site comment internally on service project issues:

1. On the project permission scheme page, select **Actions > Edit permissions**.
2. Under **Comment Permissions**, select **Edit** in the **Add comments** entry.
3. In the **Granted to** selection, choose **Application access**.
4. From the dropdown, select **Any logged in user**.
5. Select **Grant**.

Agents, not customers, will see these comments. Only Jira Service Management agents and admins can comment directly to customers. Other Jira users may comment publicly if the agent adds them as request participants.

Share your development teams' custom fields

Bug issues in Jira Software and Jira Service Management can stay in sync if they share custom fields and screens. We suggest you replicate the issue fields and screens from your development teams' bug issues in your Jira Service Management request types.

Work with your development teams' and Jira administrators to share custom fields between your request types and development projects.

[Read more about custom fields.](#)

Collect effective bug reports from customers

Jira Software is an issue and bug tracker that helps developers plan, build and ship their work. Jira Service Management is a simple way for customers to send your team bugs and feedback about shipped software.

Your users are your best friends. Their bug reports can help you:

- resolve issues before other users encounter them
- investigate and fix problems that slip through QA or automated testing
- catch issues from platforms you don't prioritize during testing
- show your users you care about quality and their experience

Issues can happen in any part of software development, not just the code. If your designers, documentation writers, product managers, or other team members track their work in Jira, you can escalate bugs related to their work directly to their teams, too.

The bug reporting and resolution process

The information needed to replicate and resolve bugs varies. But, you can standardize your process for collecting bug reports.

The customer service template associates certain requests with a bug report workflow. This workflow complements the bug report process. Use it as a jumping off point for your service project.

Requests that follow a bug report process have the same workflow in Jira Service Management and Jira Software. But, the teams' processes for handling bug reports have significant differences.

Bug process for agents working in Jira Service Management	Bug process for developers working in Jira Software
<ol style="list-style-type: none">1. A user reports a problem with their software or service.2. A service project agent investigates to see if they can provide any known solutions to the customer:<ul style="list-style-type: none">• If the problem has a known solution, the service project agent works with the customer to fix the problem.• The problem may be new or the service project agent can't find an answer from a knowledge base or development point of contact. In this case, the service project agent escalates the bug report to the development team for fixing. Read more about escalating issues to other Jira teams.3. The service project agent liaises between the customer and development team to collect any extra information needed to fix the bug. They communicate the development team's progress.4. The service project agent verifies with the customer that they fixed the problem. Then, the agent resolves the customer's bug report.	<ol style="list-style-type: none">1. The development team receives a bug report.2. The project manager determines the bug's priority. They assign the issue to a developer.3. The developer investigates the bug and either verifies or rejects the report.4. The developer fixes the problem and transitions the issue to a QA tester.5. The tester verifies they fixed problem and resolves the ticket. They notify the service project agent that development is complete. They may provide extra information, like when the fix will reach the customer.

Collect specific information from users with custom fields

Work with your development team and Jira administrators to share a custom field set. Define what custom fields you want to collect to aid developers fixing bugs. Your Jira administrator can maintain these fields in a single screen scheme. They can apply the scheme to both development and service projects. [Read more about custom fields and screen schemes.](#)

Developers note incomplete information as the biggest blocker to investigating and fixing bugs. The most common information that developers use are:

- the steps to reproduce
- observed and expected behavior
- screenshots

You might want to collect other information to categorize, report, or automate actions related to the bug report. For example:


- operating system
- version
- component
- URL
- user agent string

The more information you can collect, the easier diagnosing the problem will be. Your development team will thank you.

By default, the **Report a bug** request type comes with these fields:

- Summary
- Symptom
- Attachment

To add pre-created custom fields to your request types:

1. In your service desk project, select **Project settings** () > **Request types**.
2. In the **Report a bug** request type entry, select **Edit fields**.
3. Select **Add a field**.

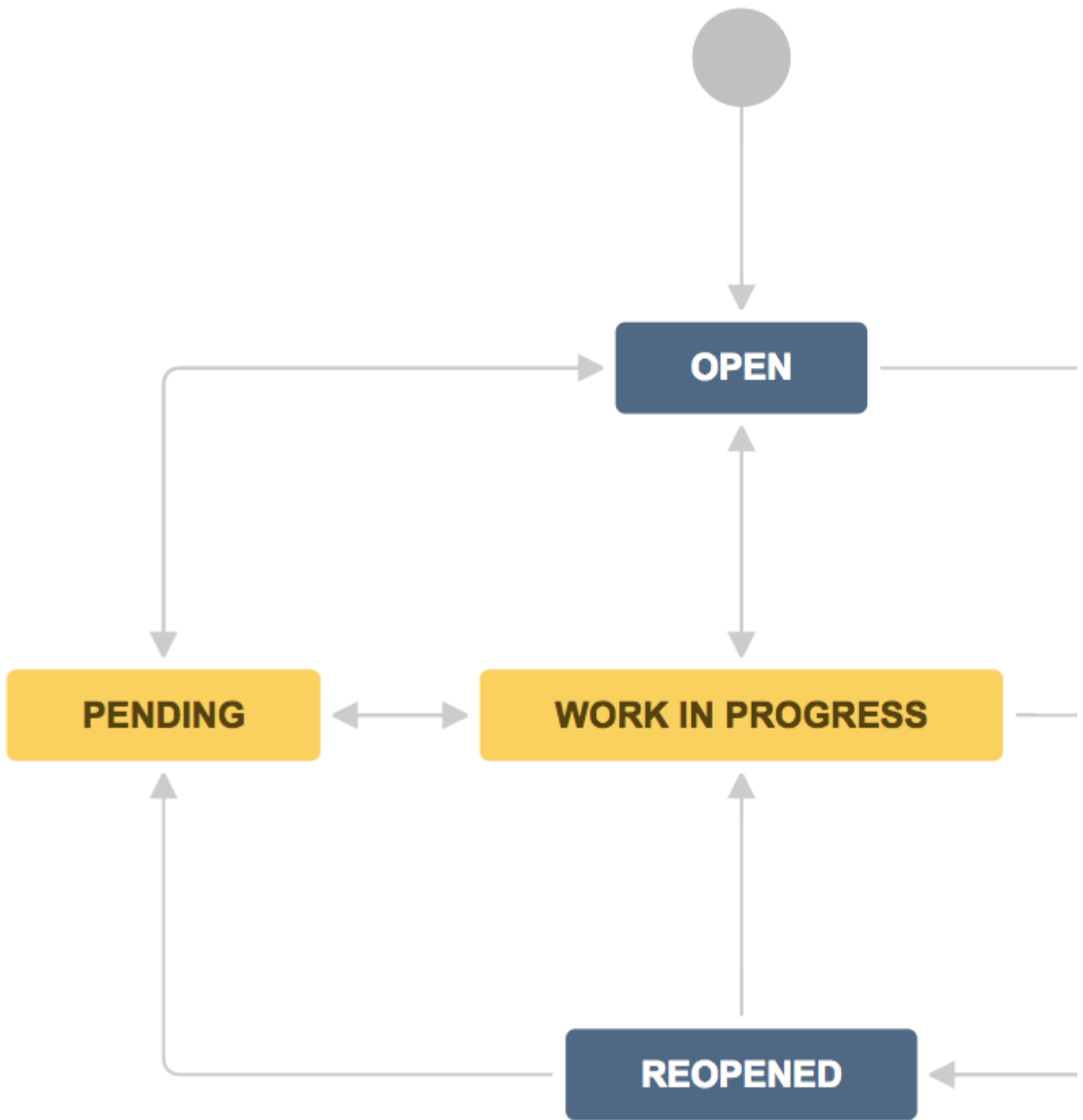
[Read more about adding custom fields.](#)

Tips for creating bug report forms on your portal

- You can add help and instructions to your bug report request type. Encourage your customers to report each problem on a separate request. This helps tracking bugs in reports and development sprints.
- Use natural language when asking for information. For example, if you add a field to collect expected behavior, ask your customer to report these in plain terms: "What did you expect to happen?"
- Check in with your development teams every so often. Ask if there's any more information they need to squash bugs in your software.
- Check in with your Jira administrators every so often. Ask if there's any changes to screens that you should reproduce in your request types.

Customize Jira Service Management's bug report workflow


When a customer reports a bug using the portal, Jira Service Management assigns the request to your organization's **Bug** workflow. By default, this workflow is the same for both Jira Service Management and Jira Software issues.



This workflow follows the basic process above. Jira administrators can customize it to adapt to the needs of your customers, developers, and service desk agents.

Customize your bug report and resolution workflow

Jira's default **Bug** workflow is a good starting point for most teams. As your organization grows, you may want to update your workflow to suit the specific needs of your business. For example, you can add steps for approvals or QA verification.

 You need to be a Jira admin to make changes to workflows. Changes to workflows affect all projects that use the workflow. Be careful!

To edit the bug workflow:

1. In your service project, select **Project settings** () > **Workflows**.
2. Select the edit icon () next to the entry titled **<Project key>: Jira Service Desk default workflow**.
3. Use the workflow editor to add or remove steps and transitions.

[Read more about working with workflows.](#)

Add a "waiting on development" step to the request's workflow

You may want to add a stage to your bug workflow for when a development team investigates the issue. This status helps your service team track which bugs are being handled by a development team. And, it informs customers when the issue is being looked at by developers.

To add a new step in the workflow:

1. Follow the above steps to customize the **<Project key>: Jira Service Desk default workflow**.
2. Select **+ Add status**.
3. Give the status a name. For example, *Waiting on development team* and select **Add**.
4. Give the status a description. This helps administrators who manage workflows across Jira Server apps.
5. Switch the category to **In progress**.
6. Select **Create**.

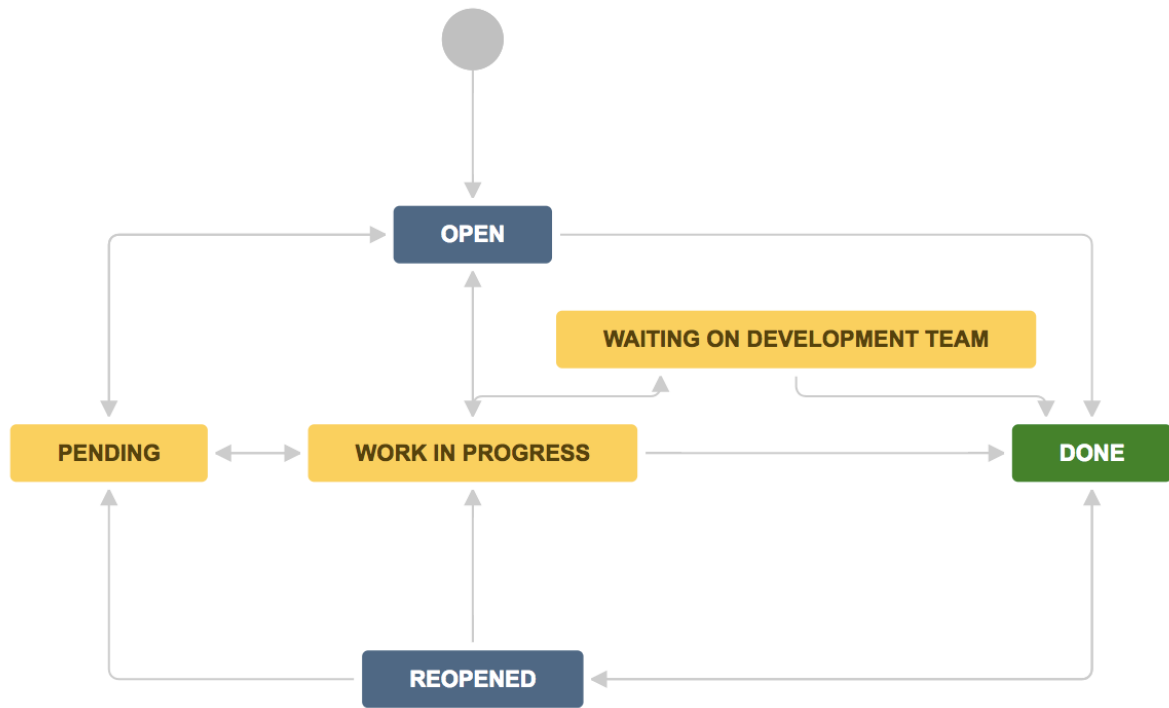
To add an incoming "escalate to development team" transition:

1. Select the **Waiting on development team** status.
2. Select **Add Transition**.
3. In the **From status** drop down, select **Work in progress**.
4. In the **To status** drop down, select **Waiting on development team**.
5. In the **Name** field, provide the call to action. For example, *Escalate to development team*.
6. In the **Screen** drop down, select **Workflow Screen**.
7. Select **Add**.

To add a transition to allow the agent to mark the request as done:

1. Select **+ Add transition**.
2. In the From status drop down, select **Waiting on development team**.
3. In the To status drop down, select **Done**.
4. In the **Name** field, provide the call to action. For example, *Mark as close*.
5. In the **Screen** drop down, select **Resolve Issue Screen**.

Publish your draft workflow.



Escalate Jira Service Management issues to other Jira teams

As agents investigate bug reports, they may need to escalate issues to the development team. For example, the agent may not find a fix from the knowledge base or a development contact. In this case, they may escalate the issue to a development team to fix in the software.

The development team should use an issue on their software board or backlog to track the fix. To make this easier, Jira Service Management agents can create these issues for them. Here's how the process works:

1. A service project agent receives a bug report from a customer.
2. The agent verifies that the bug needs a developer to fix the issue.
3. The service project agent creates a linked issue in the development team's Jira Software project.
4. The development team lead prioritizes and assigns the software issue to a developer to fix.
5. The developer works in their Jira Software project while fixing the issue.
6. The developer resolves the issue and Jira Service Management automatically updates the service agent about the fix.

Agents can pass on feature requests or improvement suggestions to project planning teams. They can also create issues with public-facing websites for marketing or documentation teams that use Jira Core.

Create linked issues

Service project agents can escalate a Jira Service Management issue to a Jira Software or Jira Core project by creating a linked issue. To create a linked issue:

1. View the issue.
2. Select the **More** option.
3. Select **Create linked issue**.
4. Choose the appropriate project where the issue needs to be escalated to.
5. Select **Create**.

Linked issues created this way use a "causes" link by default. This means:

- The service desk issue is "caused by" the linked development issue.
- The development issue "causes" the linked service desk issue.



Jira Software and Jira Core teams get the most use of this technique when they can view and comment on issues in your service project. [Read more about collaborating with other Jira teams on Jira Service Management issues.](#)

Automatically update Jira Service Management agents about the progress of a linked issue

Service projects come with an automation rule that updates agents on the status of linked issues. The rule adds an internal comment on the service desk issue whenever another team transitions a "caused by" linked issue.

So, when a developer looks into a bug and resolves it as "fixed", they notify the service desk agent in the process. Then, the agent can follow up with the customer to make sure the fix works for them.

We enable the **Update when a linked issue changes** automation rule by default. To view or edit this rule:

In your service desk project, select **Project settings** () > **Automation**.

[Read more about automation.](#)

Get set up for customer service

Create a customer service project

You need to be a Jira administrator to do this step.

To create a project using the customer service template:

1. Select **Projects > Create project**.
2. Choose the **Customer service** template and hit **Next**.
3. Name your project.
4. Select **Submit**.

Your new project comes with a service catalog, recommended workflows, and basic reports. If you bundle Jira Service Management with Confluence, we also create a Confluence knowledge base. We recommend using this to record known solutions, FAQs and other articles that your customers can use to serve themselves. [Try Confluence for free](#).

Create forms to collect the right information from your users

Our web forms make sure users give you all the details you need to give support or collect feedback. If you get all the information up front, you can:

- prioritize requests
- get them to the right agent
- give development teams the information they need to investigate and fix bugs quickly

We call these customizable web forms *request types*. To start, find the most common and urgent requests that users have sent your way, then make sure there's a corresponding request type for each of these.

To view or edit your request forms:

1. In your service project, select **Project settings** () > **Request types**. The forms there display in your web portal.

The customer service template comes with the following request types:

- Technical support
- Licensing and billing questions
- Product trial questions
- Other questions
- Report a bug
- Suggest a new feature
- Suggest improvement

[Read more about request types](#).

Fill a knowledge base so customers can help themselves

A knowledge base stores how-tos, FAQs, and other short articles. Customers can use these to solve problems without contacting support. And, agents can use these to share knowledge and solve requests faster.

The easiest way to add a knowledge base to your service desk is to link your project to a Confluence knowledge base space. When you add a knowledge base to your service project, it benefits both your users and your agents:

1. Customers search and view relevant articles when they search the customer portal. They may find the answers they need without ever raising a request.
2. Agents find relevant articles when they work on issues. This saves time finding and writing common answers or onboarding new team members.

Learn more at [Set up a knowledge base for self-service.](#)

Getting help with Jira Service Management

How can we help you?

We have a number of [help resources](#) available. You can get your problem resolved faster by using the appropriate resource.

[I don't know how to do something](#)

[Something isn't working](#)

[I don't like the way something works](#)

[Something else?](#)

I don't know how to do something

1. Search the [Atlassian Community](#).
2. Raise a [support request](#)*.

Something isn't working

1. Check the 'Jira Service Management' (previously Jira Service Desk) knowledge base at <https://confluence.atlassian.com/display/JiraKB>.
2. Search the [Atlassian Community](#).
3. Raise a [support request](#)*.

If you've identified a bug but don't need further assistance, raise a [bug report](https://Jira.atlassian.com) (<https://Jira.atlassian.com>).

I don't like the way something works

- Raise a [suggestion](https://Jira.atlassian.com) (<https://Jira.atlassian.com>).

Something else?

- If you need help with something else, raise a [support request](#)*.

* *Tip: If you are the **Jira system administrator**, you have a number of additional support tools available. See [Raising support requests as an administrator](#) for details.*

About our help resources

Atlassian Support

Our support team handles support requests that are raised in our [support system](#). You need to log in using your [Atlassian account](#) before you can raise support requests.

For information on our general support policies, including support availability, SLAs, bugfixes, and more, see [Atlassian Support Offerings](#). Note, you'll find anything security-related at [Security@ Atlassian](#).

Tip:

If you are a Jira system administrator, you have a number of additional support tools available. These include the ability to raise a support request from within your Jira applications, create zip files of key Jira application information, and more. For details, see the [Administering Jira Applications](#) documentation.

Atlassian Community

[Atlassian Community](#) is our official application forum. Atlassian staff and Atlassian users contribute questions and answers to this site.

You may be able to find an answer immediately on the Atlassian Community, instead of having to raise a support request. This is also your best avenue for help if:

- you are using an unsupported instance or an unsupported platform,
- you are trying to perform an unsupported operation, or
- you are developing an app for Jira Service Management.

You can also have a look at the [most popular Jira Service Management answers](#) and the [most popular Jira answers](#).

Jira Service Management knowledge base

If there are known issues with a version after it has been released, the problems will be documented as articles in our [knowledge base](#).

Atlassian issue tracker

Our official [issue tracker](#) records our backlog of bugs, suggestions, and other changes. This is open for the public to see. If you log in with your Atlassian account, you will be able to create issues, comment on issues, vote on issues, watch issues, and more.

**Tip:**

Before you create an issue, search the existing issues to see if a similar issue has already been created.

Automate your project

i Automation is now part of Jira Service Management Data Center. To start using it in your projects, upgrade to Jira Service Management Data Center 5.0 or later. Can't upgrade yet? Download the equivalent [Marketplace app](#) free of charge.

Automation lets you set up rules to automate repetitive tasks around Jira, such as closing stale issues, auto-assigning issues to the right people, or sending notifications to your teammates.

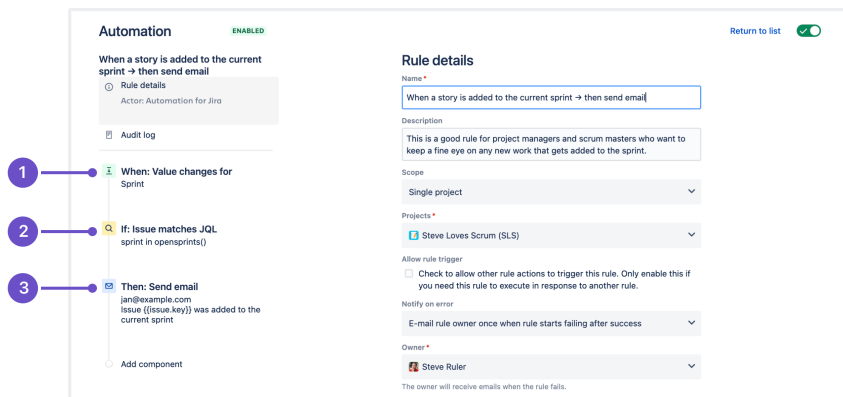
Create a rule

Each rule is made up of a trigger that kicks off a rule, conditions that refine the rule, and actions which execute it.

To create a rule:

1. In your project, go to **Project settings**.
2. Select **Automation**.
3. Select **Create rule**. You'll then add triggers, conditions, and actions.

Read more about the rule components below.



1. Triggers

Every rule starts with a trigger. You can choose from many triggers including issue transitioned, scheduled trigger, or issue created. If you wish, you can customize your trigger before moving on to your next component and completing your rule. [Learn more about triggers](#)

2. Conditions

Conditions allow you to narrow the scope of the rule. For example, only escalate an issue if it's high priority. Rules don't have to have conditions, but if you want to include them, they can be placed anywhere in the rule chain. If one condition fails, the rule will stop running. [Learn more about conditions](#)

3. Actions

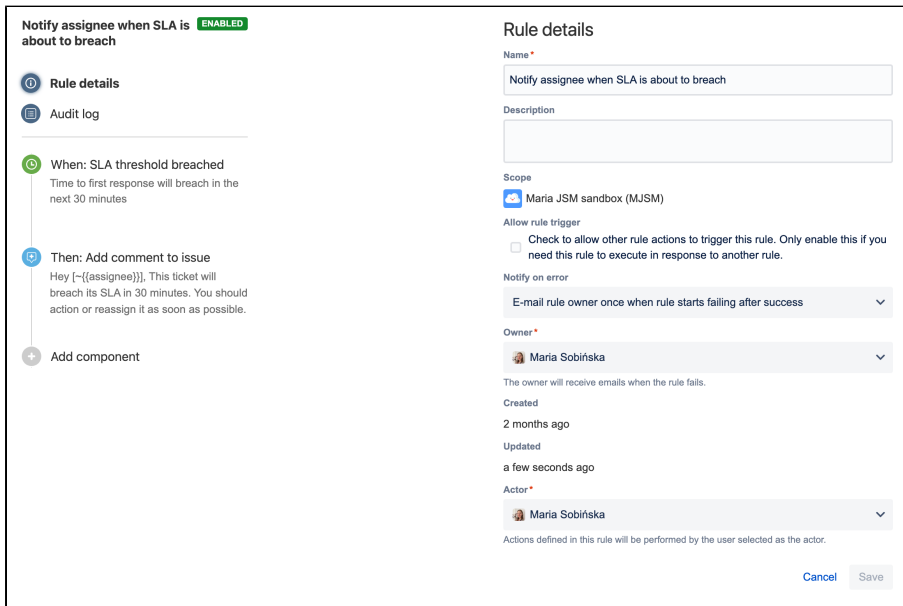
Actions run once a rule is triggered and all conditions are met. You can perform many tasks, such as editing issues, sending notifications, creating sub-tasks, or adding audit messages. [Learn more about actions](#)

Use automation at scale

To give you an overview of what you can achieve with automation, we've prepared three examples of rules you can use to make the work go smoother.

Keep response times on track

This rule reduces the number of times you need to go over your SLAs by notifying the assignee that they need to take an action when an SLA threshold is about to be breached. This way you get more time to focus on other things while your customers receive a response to their requests on time.

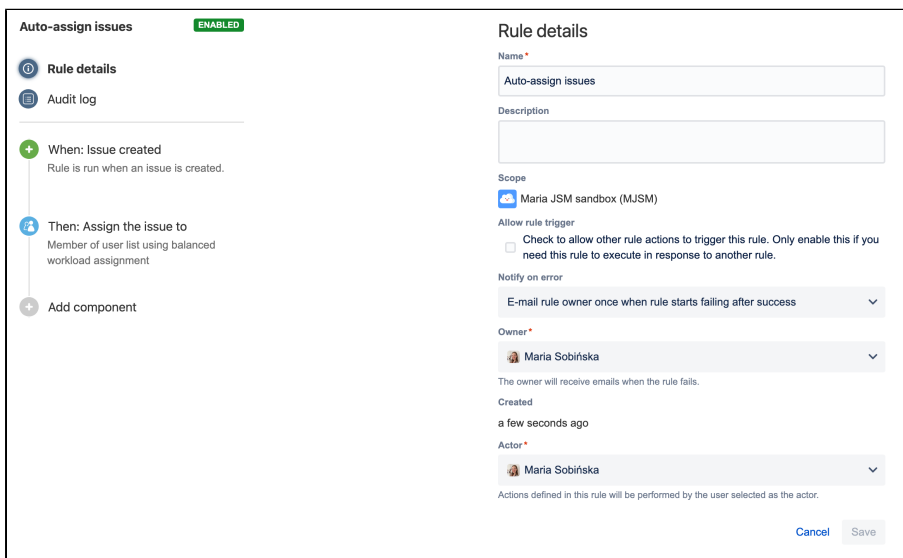


The rule consists of:

- A trigger: An SLA threshold will be breached in the next 30 minutes.
- An action: Add a comment to the issue notifying the assignee about the breach.

Balance the workload

Ensure that your team shares the workload so you can respond as quickly and effectively as possible. Whenever an issue is created, this rule searches for a member of your team with the least amount of open tickets.




The rule consists of:


- A trigger: An issue is created.
- An action: Assign the issue to a team member using the balanced workload assignment method.


Improve customer communication

Remind your customer to respond. Close the issue when they dont. Then, reopen it when they do. Simplify your communication with customers with these three rules.

Remind your customer to respond

 Rule details


 **When: Scheduled**
Searches with:
status = "Waiting for customer" and updated < -2d
every 1 Days


 **Then: Add comment to issue**
Hi {{reporter.displayName}}, [something about needing the customer to respond]


The rule consists of:

- A trigger: A scheduled daily search for issues with the Waiting for customer status that havent been updated in 2 days.
- An action: Add a comment notifying the reporter they need to remind the customer to respond.

Resolve an issue due to inactivity

 **When: Scheduled**
Searches with:
status = "In Progress" and updated < -5d
every 1 Days

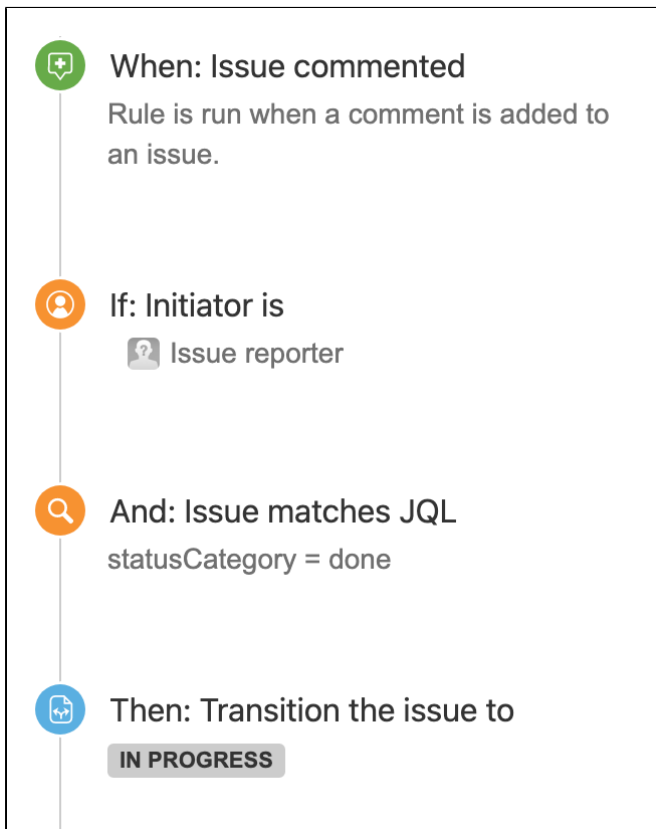
 **Then: Transition the issue to**
RESOLVED

 **And: Add comment to issue**
Hi {{issue.reporter.displayName.split("\").first}}, We haven't heard back from you for 5 days, so we're automatically

The rule consists of:

- A trigger: A scheduled daily search for issues with the In progress status that haven't been updated in 5 days.
- Actions: Change the status to Resolved. Add a comment notifying the reporter the issue is being automatically closed due to inactivity.

Reopen an issue when there's a new response




The rule consists of:

- A trigger: A new comment is added to the issue.
- Conditions: The comment was added by the issue reporter. The issue is closed.
- Actions: Change the status to In progress.

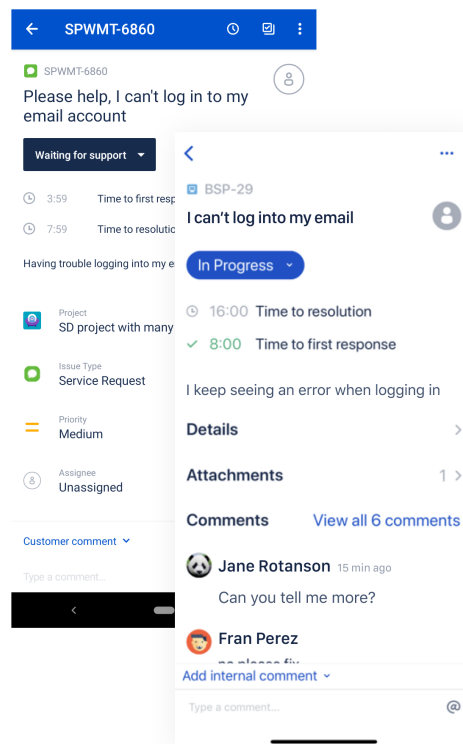
Get started with Jira automation

To get started, see [Jira automation](#).

Jira Data Center and Server mobile app

 Jira Service Management has successfully passed the Beta stage and is now officially supported in the mobile app. To get this version, you'll need to upgrade to Jira Service Management 4.15 or later. If you can't upgrade yet, you can continue using Beta.

A native app for Jira Service Management Data Center and Server is here!



About the app

Here's what you'll get, and what you can do in the app.

Available right now

- Viewing Jira Service Management projects and requests
- Commenting internally and externally
- Notifications about request updates
- Working on requests (creating, editing, linking, reassigning, responding to customers, etc.)
- Viewing SLAs and planning which tasks need to be resolved first
- Jira Service Management specific fields (Organization, Request type, Request participant), view-only

Still working on

- Request support (Approvals, attachments)
- Allowing editing of Jira Service Management specific fields in the app
- Agent queues

Requirements

Jira Service Management

You'll need to install the latest version of the [Mobile Plugin for Jira](#) from Atlassian Marketplace, or upgrade to Jira Service Management 4.15 or later, which includes the plugin out of the box.

Device requirements

To use the app, your users will need a device with either:

- Android 5.1 (Lollipop) or later, or
- iOS 14 or later (iPhone, iPad, or iPod Touch)

Downloading the app

If you already used our Jira Data Center and Server app for Jira Software or Jira Core, you're good to go, it's the same app. If not, you can download it for [Android](#) or [iOS](#).

Considerations for administrators

Here are some things to consider when determining whether your users will be able to use the app.

VPN and firewalls

If your Jira instance is not accessible on the public internet, users will need to connect their device to your network or virtual private network (VPN) in order to use the app.

We recommend providing your users with step-by-step instructions on how to connect to your VPN when you let them know the mobile app is available, as this is something Atlassian Support will not be able to help them with.

SSL

The app accepts both HTTP and HTTPS connections.

If your Jira instance is configured to use SSL, you will not be able to log in on the app if:

- your certificate is self signed
- the Certificate Authority (CA) is unknown, or is not one that Android / iOS trusts by default (for example it might be a new CA that is not yet trusted, or a private CA)
- your certificate is missing an intermediate CA, affecting the certificate chain
- your certificate doesn't meet Apple's [Requirements for trusted certificates in iOS 13](#) (affects people using the app on iOS devices).

See our [Knowledge base article](#) for information on how to resolve this.

Storage and encryption

The iOS and Android apps cache some content (issues, projects, boards) locally on the device. This helps keep the app responsive when navigating around projects and issues. We don't use any application-level encryption when storing cached data, but the device's internal storage may be encrypted by the operating system.

When a user logs out, all cached data is deleted.

We don't store passwords in the app. Instead we use session cookies, which are encrypted by default.

Login and authentication

The app supports all common Jira user management configurations, including external user directories and SAML single sign-on (NTLM method isn't supported, only basic). Users will need to sign in to use the app, even if your site allows anonymous access.

The app uses an [RFC-compliant](#) user-agent header for making the requests during login. This allows us to effectively identify the app and apply further processing based on the users hardware and software.

Given the wide variety of supported configurations we decided to adopt the most versatile format. For both iOS and Android we compose the User-Agent header value by concatenating the standard WebView User-Agent and unique app identifier.

iOS User-Agent

Format
Mozilla/5.0 (<system-information>) <platform> (<platform-details>) Safari/<technical-version> AtlassianMobileApp
Example
Mozilla/5.0 (iPhone; CPU iPhone OS 14_7 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Safari/604.1 AtlassianMobileApp

Android User-Agent

Format
Mozilla/5.0 (<system-information>) <platform> (<platform-details>) AtlassianMobileApp
Example
Mozilla/5.0 (Linux; Android 9; Mi A1 Build/PKQ1.180917.001) AppleWebKit/537.36 (KHTML, like Gecko) AtlassianMobileApp

These headers are used in iOS app version 1.27.0 and Android app version 0.21.6 and later.

Mobile Device Management (MDM)


You can distribute the Jira Server app to people in your organisation using your MDM solution. For more info on how to do this, see [Mobile Device Management](#).

Third party add-ons and visual customizations

The mobile app provides a simple, lightweight way for users to view, create, edit and collaborate on issues. Complex interactions, including those provided by add-ons, will not be available in the app.

Any look and feel customizations youve made to your Jira instance will not be reflected in the app.

The Jira Data Center and Server mobile app can push notifications directly to users' devices. Users choose whether they'd like to receive push notifications from the app, and can opt out at any time. This feature uses a cloud-based notifications service developed and maintained by Atlassian and hosted on our AWS infrastructure. No user or message content is sent to the service, only notification IDs, and we don't store any data.

- If you need to avoid using any cloud-based services you can choose to disable push notifications entirely. Head to **Administration** () > **System** > **Jira mobile app** .
- If you're using restrictive firewall or proxy server settings, you'll need to allow (whitelist) <https://mobile-server-push-notification.atlassian.com> to ensure push notifications work as expected.

For sites that are not accessible on the public internet (for example users need to be connected via VPN to use the app) we adapt the push notification message as follows:

- If the user is connected to your network or VPN, we'll show the full notification, for example "Sara Leung shared 'End of year party' with you"
- If the user is not currently connected to your network or VPN, we'll show a shorter notification, for example "1 new notification".

For more info about push notifications, see [Push notifications](#).

Related pages and known issues

If youre having problems with logging in or encountered some other issues, head on to our knowledge base articles that will tell you what to do:

- [Jira Data Center and Server mobile apps](#)

Accessibility

We want every team around the globe to be able to use Jira with the least amount of trouble, and the accessibility settings bring us closer towards this goal. Whether your vision is impaired, you can't really tell colors apart, or just strongly believe that blue, azure, and sapphire are the same thing we've got you covered.

More accessibility improvements!

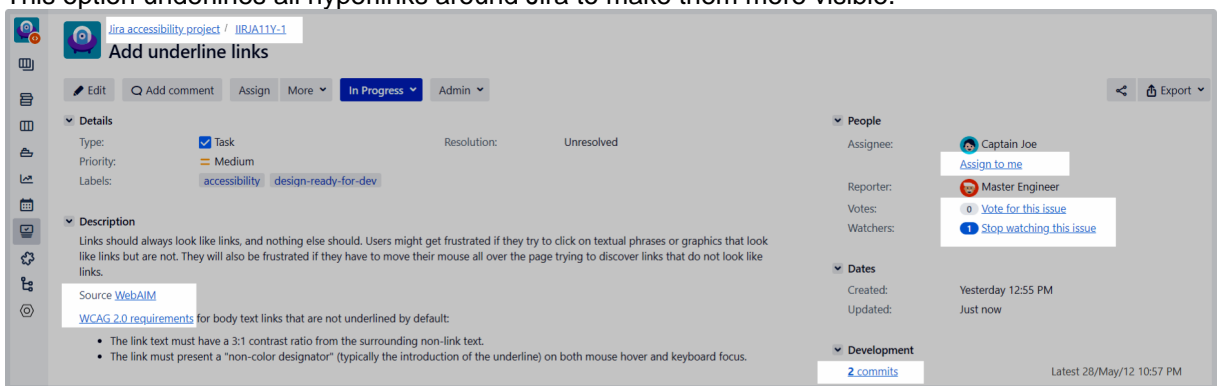
Accessibility in Jira is not a one-time effort we're constantly reviewing our VPAT report and fix accessibility issues in almost every Jira version. For more info, see [Accessibility improvements in Jira](#).

Changing your accessibility settings

You can personalize your accessibility settings to make it easier to work with Jira. To change the settings, click your user avatar and select **Accessibility**. You can choose the following options:

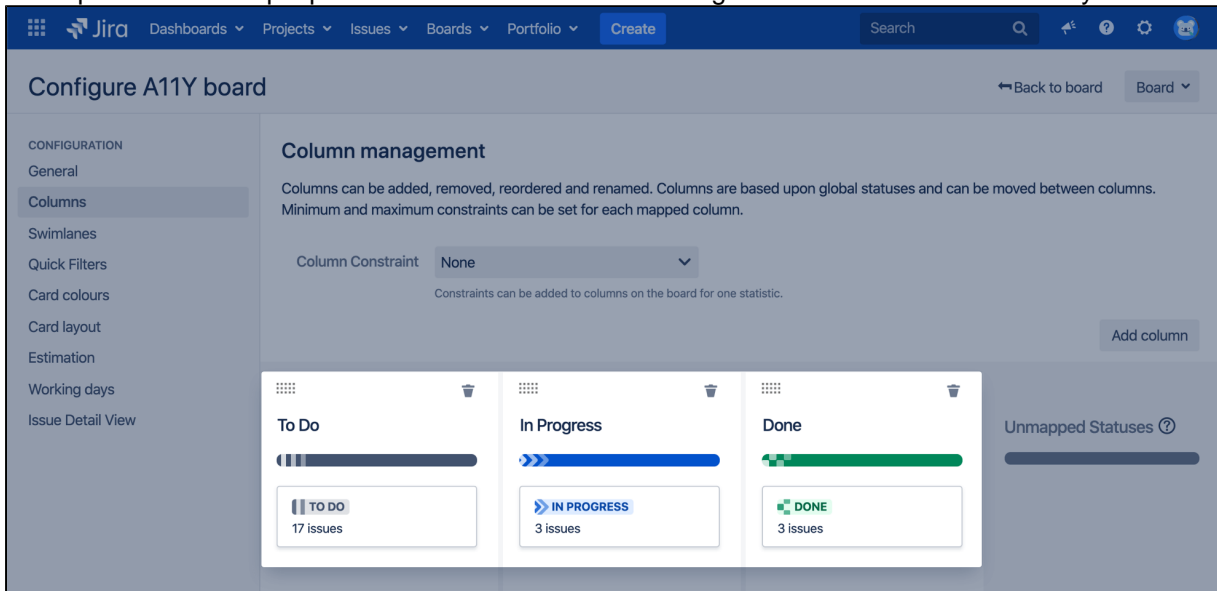
■ Underlined links

This option underlines all hyperlinks around Jira to make them more visible.



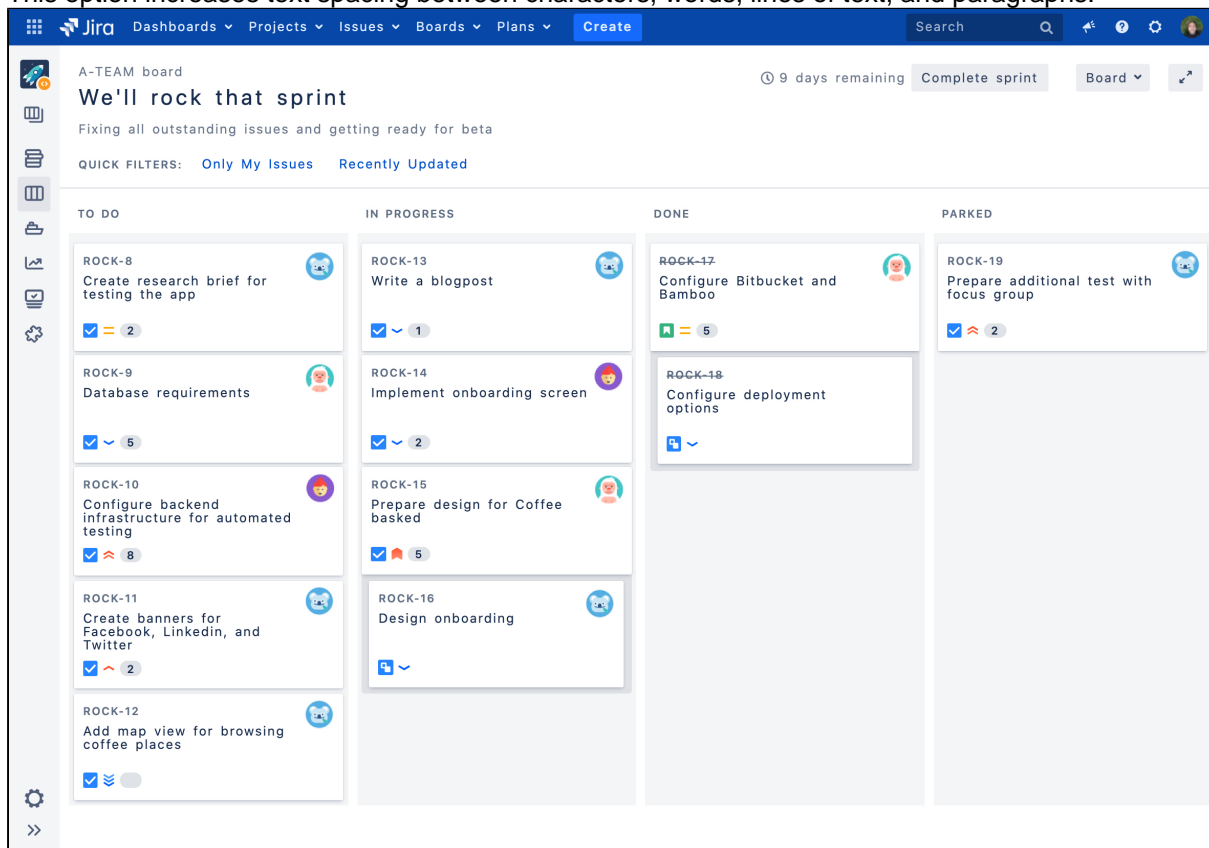
■ Patterns on issue statuses

This options adds unique patterns to issue statuses to distinguish between them more easily.



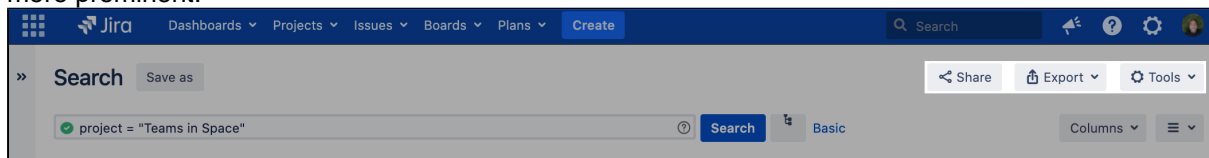
■ **Increased text spacing**

This option increases text spacing between characters, words, lines of text, and paragraphs.



■ **Background in subtle buttons**

This options adds a gray background to subtle buttons (normally displayed on hover) to make them more prominent.



For app developers

If youre an app developer, you can make your app add new options to accessibility settings. For more info, see [Accessibility for app developers](#).