ATLASSIAN

Jira Core Server 9.10

Contents

Jira Core Server 9.10 documentation

Jira Core is a customizable workflow solution that helps you manage your projects and keep your team organized.

Get started

New to Jira Core? 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 Core 9.10.

View latest changes

Jira Core overview

A project and task
management solution
for business teams

Resources

Topics

- Jira Core tips
- How to build a workflow
- What makes up Jira Core?
- Jira Core use cases
- Searching for issues
- Managing project roles

**Blogs**

- Say hello to Jira Core!
- How to set up business workflows
- Make issues work for your team
- Project status at a glance
- Helping Marketing and Software teams work together
- 3 steps to Marketing zen

What makes up Jira Core?

Jira Coreis a Jiraapplication that provides you with a workflow management system that you can use for many things, including running projects, tracking assets, and basically anything that requires work moving through a workflow.Jira Corecan be customized to suit your needs, and can also be extended and linked with other applications to provide you with the perfect solution to track all your work in one place.

Understanding the basics ofJira Corewill help you get the most out of your application. This page will explain the core terminology and functionality that makesJira Corethe application you need to help you manage your work efficiently.
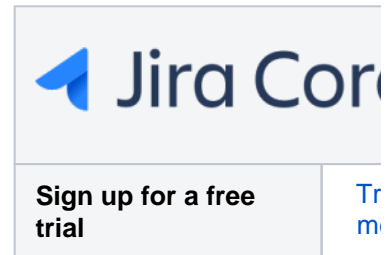
Ready to get to grips with the basic concepts? Read on!

Issues

Issues are the work packets inJira Core. Each issue can be further defined by assigning the issue an issue type. For example, if you're running a project in an office, issues could represent the tasks you need to do to complete that project. Each issue type would be a type of task, like administration task, filing task, or create document task. If you're usingJira Corefor asset tracking, an issue could represent an asset (or inventory item) and the issue type could be the types of assets (laptops, monitors, printers etc.). Issues then progress (or move) throughJira Corevia an associated workflow that dictates what can and can't (or more correctly, what should and shouldn't!) happen to that issue.

So how do I group issues? What if I want a construction project, and I also want to track my assets for that project inJira Core, how can I do that? Well, you use projects!

Projects

Projects are a way to group your issues, and apply a set of defaults. These defaults make sure all your issues have the information that they need to be progressed and tracked through your workflow. Each project can have an administrator, who is typically the project lead, and they're responsible foradministering the project.

Jira Coreusers maywork with issuesin one or several projects, and would complete the work required to progress the issues through their workflows.

Great! You now know issues are the work, and they're grouped in projects. But what's the workflow, and how does it effect the issues?

Workflows

Workflow dictates how an issue can be progressed in a project. The workflows can be as simple or as complex as you need them. Workflows are often modeled on existing processes, and are made up of statuses (or steps) and transitions (movements between statuses). When you create an issue, it will automatically be assigned a workflow and a status on that workflow. Where it can move to is defined by the transitions that exit that status. An example of the default workflow that ships withJira Coreis shown below:

WORKFLOW

Jira applications

Jira Coreis one of several Jira applications. Each application has its own distinct features that makes it suitable for servicing the market it's designed for. For example, Jira Software is designed for agile software developers, and its features include agile boards, software specific project types, and better integration points with other developer tools.

You may have access to more than just theJira Coreapplication, and you can viewthis list of applications and feature access to gain a better understanding of the power of Jira!

So that's the basics.Jira Coreis functional straight 'out of the box', but if you think you'll need to know a little more, to get some ideas on what you could do, take a look at this guide tosee what's possible with Jira Core.

See what's possible with Jira Core

Jira Corecomes ready to use right out of the box, and the project templates are a great way to get your first project up and running. In this section of the documentation, we'll take you through the project templates, and give you some ideas of how you may want to customize your projects so that they suit your needs. We'll also give you some tips on how to get the most out of your projects and issues.

Jira Core

Sign up for a free trial | Try me!

Task management
The Task management template sets you up with a simple workflow, easy to understand issue types, and the right set of fields on your issues to help you get your work from To Do to Done in the quickest way possible.

Learn more about the default configuration and specific use cases...

Project management
The Project management template has a slightly more complex workflow, and is great for projects that require a little more work. Including an In Progress status allows you to work on tasks over longer periods and show work in progress.

Learn more about the default configuration and specific use cases...

Process
management
The Process management template ships with our most complex workflow, designed to give you an idea of what you can create with Jira Core to mimic your own business process or work flow.

Learn more about the default configuration and specific use cases...

HR
If you're working in HR, you may want to have a look at a few ideas of how you can use Jira Core for your projects, and configure them further to suit your needs.

Learn more about using Jira Core for HR...

Marketing
If you're a marketing whiz and looking to move to the next level by managing your work more efficiently, making sure all your I's are dotted and T's crossed, have a look at how you can use Jira Core to help you work smarter and more effectively.

Learn more about using Jira Core for marketing...

Operations
Thinking about trying to set up some workflows for operations? Wondering how Jira Core could help you keep things in one place while still letting you apply the workflow you need? Take a look at our operations example for some ideas.

Learn more about using Jira Core for operations...

Finance
If you work in finance and struggle to keep tabs on everything required for reporting, budgeting, invoicing.... you know what you do better than we do! Have a look at how you could potentially leverage Jira Core to make sure all your reports are done on time, all your invoices are issued and followed up, and you can check up on where you stand at any given time.

Learn more about using Jira Core for finance...

Legal
Keeping tabs on everything you're working on can be hard, and when you also need to make sure things are reviewed and approved by the relevant parties, things can get harder to manage. Jira Core can help you manage your work and keep tabs on it at each stage through the workflow.

Learn more about using Jira Core for legal...

Task management

The task management project template sets you up with the most basic workflows, and is designed to allow you to create issues and get them completed with the minimal of fuss. UsingJira Core's functionality you can involve the members of your team that you need to, and track your work using your built inreporting, yourdashboard, andcustom filters based on your searches.

Sign up for a free trial	Tr mo	
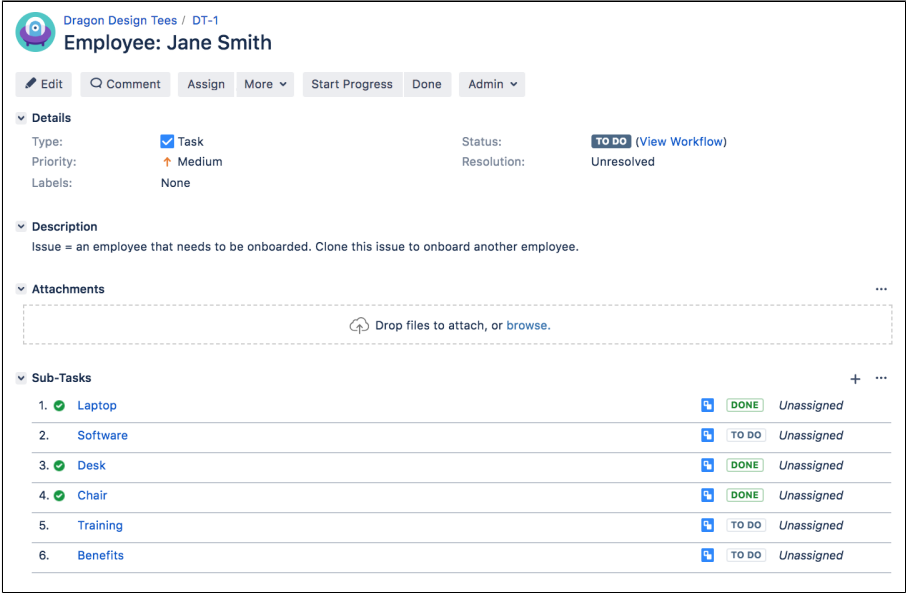
Out of the box, your task management project comes with:

Workflow	To Do > Done WORKFLOW TO DO → DONE
Issue types	Task and sub-task. A sub-task must belong to a parent task. ISSUE TYPES ☑ Task ⊞ Sub-task
Issue fields	Summary, Issue Type, Reporter, Attachment, Due Date, Description, Assignee, Priority, Resolution and Labels
Resolutions	Done, Won't Do, Duplicate and Cannot Reproduce
Priorities	Highest, High, Medium, Low and Lowest

There's a lot you can do with a task management project, and with a few customizations, there's a lot more you can do to! Have a look at the example for a few ideas of how you can use it, and customize it to suit your needs.

Use case	**Onboarding** - setting up your new starters for success from day one, week one.
What Business project should I start with?	**Task Management** - This gives you a straightforward workflow of To do and Done, and accompanying issue types of Task and Sub-task.
How can I do this?	1. Create a task for each new employee 2. Create a sub-task for everything that needs to be done before the person starts. You may need to make sure a desk and chair is available, allocate a laptop, or make sure a building access card is granted. 3. Assign each sub-task to the relevant person responsible for making sure it's done. They can complete the sub-task when they've completed their task. 4. Once all these sub-tasks have been completed, you can mark the parent task as Done! Your new employee is ready to go!

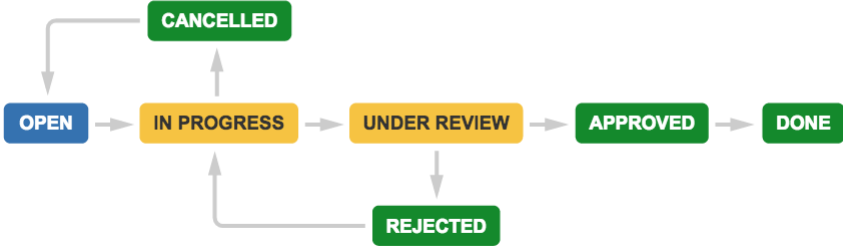|	
How would this look?	Dragon Design Tees / DT-1 **Employee: Jane Smith** ✏ Edit 🔍 Comment Assign More ▾ Start Progress Done Admin ▾ ▾ **Details** Type: ☑ Task Status: **TO DO** (View Workflow) Priority: ↑ Medium Resolution: Unresolved Labels: None ▾ **Description** Issue = an employee that needs to be onboarded. Clone this issue to onboard another employee. ▾ **Attachments** ⋯ ⤒ Drop files to attach, or browse. ▾ **Sub-Tasks** + ⋯ 1. ✅ Laptop 🔲 DONE *Unassigned* 2. Software 🔲 TO DO *Unassigned* 3. ✅ Desk 🔲 DONE *Unassigned* 4. ✅ Chair 🔲 DONE *Unassigned* 5. Training 🔲 TO DO *Unassigned* 6. Benefits 🔲 TO DO *Unassigned*
Can I make this easier?	You may be thinking, "Hold on a minute, I don't want to have to create a bunch of sub tasks every time I need to onboard someone!" Well, you don't. Set up a task with all the associated sub-tasks, and name it Onboarding. Whenever you need to onboard a new employee, find the task and use Jira Core's clone functionality to make a brand new task with all the associated sub-tasks. By renaming the task you'll know which new employee each task belongs to. You can even assign the sub-tasks to the relevant people, so that when you clone the task, Jira Core will make sure the relevant people are assigned to their tasks automatically.
Other customizations	You may decide that your onboarding should be more extensive, you may decide that instead of just getting the physical bits and pieces together for a new employee, you may want to also ensure that in their first week the get to have lunch with their team, sign off on all your statutory training, or get a building tour. You can customize your project's workflow to suit your needs, assigning the parent task to the person responsible for each step. You could decide to customize the fields available on the task, to make sure you record the information you need along the way (scores in training, asset numbers of any equipment that's been provided). You set up Jira Core to mimic how you do onboarding. It can be as complex or as basic as you like!

Process management

The process management project is an example of a project with a slightly more complex workflow that you could set up in Jira Core. The workflow shows an example of how you could create your work, progress it, have it reviewed, approved, and eventually completed. If you're interested in getting more from your workflows, you should review Working with workflows for some ideas on what you could do with your workflows to get the most out of Jira Core.

Jira Cor
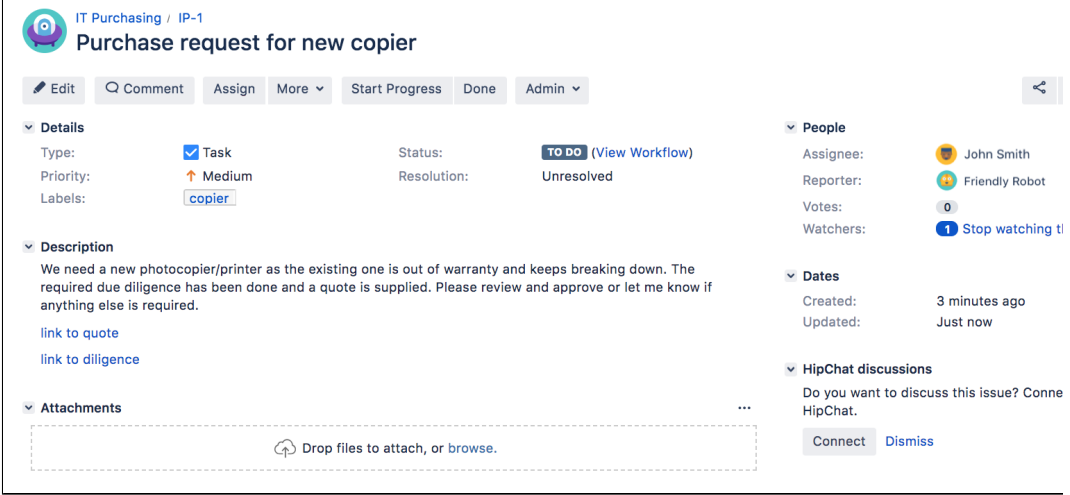
| **Sign up for a free trial** | Tr m |

Your process management project is set up by default like this:

Workflow	In a perfect world, your work would move from Open > In Progress > Under Review > Approved > Done, but there's options for the work to be returned to previous statuses via Rejected and Canceled. WORKFLOW CANCELLED OPEN → IN PROGRESS → UNDER REVIEW → APPROVED → DONE REJECTED
Issue types	Task and sub-task. A sub-task must belong to a parent task. ISSUE TYPES ☑ Task ⊞ Sub-task
Issue fields	Summary, Issue Type, Reporter, Attachment, Due Date, Original Estimate, Remaining Estimate, Description, Assignee, Priority, Resolution and Labels
Resolutions	Done, Won't Do, Duplicate and Cannot Reproduce
Priorities	Highest, High, Medium, Low and Lowest

Processes can be as complicated or as simple as you need them to be, and Jira Core allows you to achieve just that. For more ideas on what you could do, check our example for some pointers.

Use case	**Purchasing** - Creating and submitting a business case for approval
What Business project should I start with?	**Process Management** - This gives you a workflow of Open > In Progress > Under Review Approved > Done, and accompanying issue types of Task and Sub-task.

How can I do this?	1. Create a task for each new business case for approval. 2. Populate the task with all the relevant details, attaching any documents, spreadsheets additional information that you think will support the business case. 3. Moving the task to Under Review and assign it to your approver (this could be your ma or maybe a member of the purchasing department). 4. Once approved, the final stage could be to either raise a purchase request, or possibly transfer the data to another system to get the purchase completed. Either way, the issu Done and completed!
How would this look?	
Can I make this easier?	Yes! You can use a lot of Jira Core's functionality, especially comments and mentioning ot people (collaboration is alive and well!), to make sure you have all the information you nee complete the business case. Assigning the task to the relevant people at the right time as moves through the workflow makes sure it'll always appear on their to do lists, and if notific are set up, they'll even receive an email to let them know it's ready for them.
Other customizations	You may decide to change the workflow. You may need to add in a second approver. You also want to consider assigning the task automatically to the correct person as it progresse through the workflow. You may consider adding different issue types for each type of busir case that you accept, and assigning each issue type a specific, custom workflow (with the assignees per status of course!). Jira Core allows you to achieve this.

Project management

The project management project offers a slightly more robust workflow, allowing you to work on bigger items by offering a workflow that has an In Progress status. This allows you to start work on a task, and keep working in it until it's complete. Track your work inJira Coreby using our built inreporting, yourdashboard, andcustom filters based on your searches.
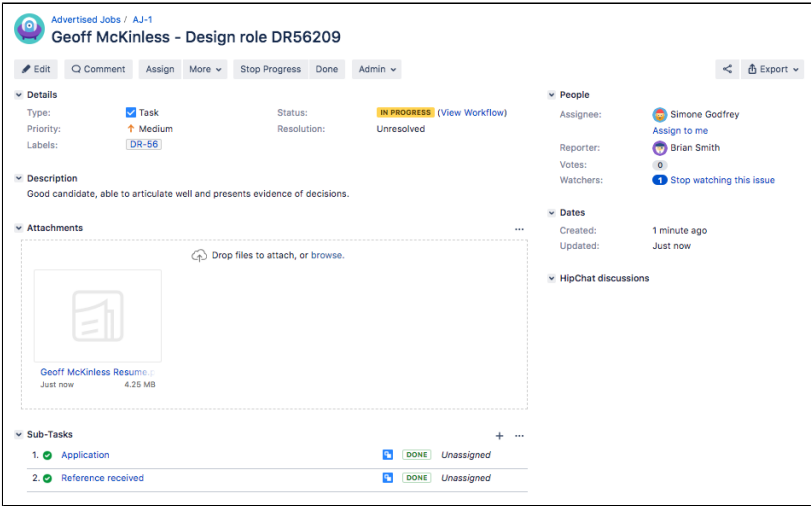
Workflow	To Do > In Progress > Done **WORKFLOW** TO DO → IN PROGRESS → DONE
Issue types	Task and sub-task. A sub-task must belong to a parent task. **ISSUE TYPES** ☑ Task 🔲 Sub-task
Issue fields	Summary, Issue type, Reporter, Attachment, Due Date, Original Estimate, Remaining Estimate, Description, Assignee, Priority, Resolution and Labels
Resolutions	Done, Won't Do, Duplicate and Cannot Reproduce
Priorities	Highest, High, Medium, Low and Lowest

Feel like project management will suit your needs, but think you may need a little more? Have a look at some of our examples for some tips and tricks on what you can do with the project management project.

Use case	**Candidate tracking** - You advertise for a role that's become available, and you start receiving applications in the mail, through your website, and via email.
What Business project should I start with?	**Project Management** - This gives you a workflow of To do, In progress and Done, and accompanying issue types of Task and Sub-task.
How can I do this?	1. Create a task for each candidate applying for a role. 2. Add any accompanying CV's, resums or paperwork as an attachment to the task. 3. Depending on your hiring process, if the candidate is unsuccessful they may go directly to Done, or if you need to perform further tasks, like back ground checks, interviews, or wait for additional paperwork, the candidate may go to In progress. 4. Once you have a successful candidate, move them to Done and add a comment or label to say they've been successful.

How would this look?	
Can I make this easier?	Yes, you can! If you've integrated Jira Core with your email system, you can make sure that incoming emails regarding job applications are automatically created as tasks. Any accompanying attachments on the email will also be added as attachments to the task. Ever wondered why some job advertisements ask for your name, the position you're applying for and job reference in the email subject line? It's because this is exactly what they're doing, taking those emails and automatically converting them to "something" they can track and process! Why not use Jira Core to do it for you.
Other customizations	We know that processing a candidate for a role is never as easy as it sounds. Depending on the role, and any requirements you may need to do several interviews, background checks, make sure you've got the correct paperwork signed and returned... the list goes on! Jira Core allows you to customize the workflow to suit this, and you can assign the task to the relevant person at each stage. Want to use the same project for all your job applications, but worried after a while it'll get too large with too many tasks open? Well you can help sort your issues by assigning them defined labels for each open role. This will help you search the tasks easily and pick out the tasks that relate to the role you're hiring for. You could also add components to the project, which are like a grouping for your tasks. You can use components to automatically assign someone to each task that's placed within that component.

Maybe you're in marketing? Have a look at how you could maybe useJira Coreto manage your work.

Use case	**Marketing campaign** - you need assets, locations booked, websites created, approval to list goes on!
What Business project should I start with?	**Project Management** - This gives you a workflow of To do, In progress and Done, and ac issue types of Task and Sub-task.
How can I do this?	1. Create a new task for each item of work that needs done. Give it a descriptive title to n everyone knows at a glance what it's for. 2. Add sub-tasks for any additional work that may need done, making sure you break big work into more manageable pieces. 3. When the sub-tasks are done, you can complete the parent task. And your work is don

How would this look?	**Adamas Marketing Campaign / AMC-1** **Submit website IA for approval** Edit Comment Assign More ∨ Stop Progress Done Admin ∨ **∨ Details** Type: ☑ Task Status: **IN PROGRESS** (View Workflow) Priority: ↑ Medium Resolution: Unresolved Labels: IA Website **∨ Description** The website information architecture needs to be submitted for approval on the 30/5/18. It must contain: • Home page • 5x sub pages (1 for each business group) • Contact page • About page • Services **∨ Attachments** ... ☁ Drop files to attach, or browse. **∨ People** Assignee: 🔴 Brian Sm Reporter: 🔴 Brian Sm Votes: 0 Watchers: 1 Stop wa **∨ Dates** Created: 2 minutes a Updated: Just now **∨ HipChat discussions** Do you want to discuss this issue Connect Dismiss
Can I make this easier?	Don't forget you can add comments to your tasks, and mention people in the comments by mention. This makes sure everyone is aware of the work and tasks that are needed. Wher and sub-tasks to track your work, it's also useful to remember you can view the Activity an page for your project by clicking on the project name in your sidebar. These views give you of the issues in your project so you always know where you stand. If you need more detail can also create you own dashboard with all the statistics you need, and add a link to the d your sidebar.
Other customizations	Marketing tasks can be very diverse and very specific in their requirements. You can custo issues and make various fields 'required' to ensure they're always completed correctly. Yo fields in various ways (drop-down, multi-select, free text etc.) to make sure whoever create can only select from your pre-defined selections, or can complete the information as they s Notifications can also be customized so that the right people are always updated when the Make sure the tasks and project are set up they way **you** work.

14

Using Jira Core for HR projects

So you're thinking about using Jira Core for your HR tasks? Maybe you're thinking about using it to progress applications? Or maybe you'd like to make sure all your new staff are set up for success by setting up a flow for onboarding? We can help you get started, show you some tweaks you can make, short cuts you can take, and customizations you might want to think about to make sure your project reflects how **you** work, not how we think you should work! We'll list out a few scenarios here, and give you some suggestions on what you may want to do to make your project a success.
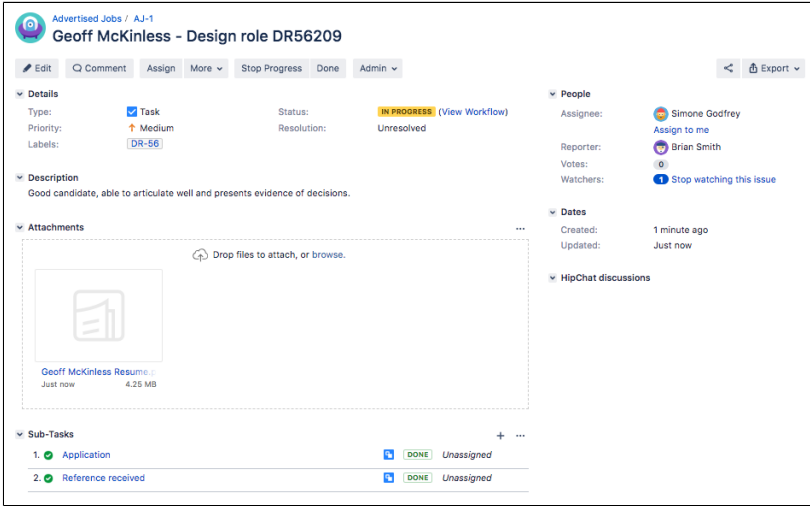
Let's look at onboarding...

Use case	**Onboarding** - setting up your new starters for success from day one, week one.
What Business project should I start with?	**Task Management** - This gives you a straightforward workflow of To do and Done, and accompanying issue types of Task and Sub-task.
How can I do this?	1. Create a task for each new employee 2. Create a sub-task for everything that needs to be done before the person starts. You may need to make sure a desk and chair is available, allocate a laptop, or make sure a building access card is granted. 3. Assign each sub-task to the relevant person responsible for making sure it's done. They can complete the sub-task when they've completed their task. 4. Once all these sub-tasks have been completed, you can mark the parent task as Done! Your new employee is ready to go!
How would this look?	
Can I make this easier?	You may be thinking, "Hold on a minute, I don't want to have to create a bunch of sub tasks every time I need to onboard someone!" Well, you don't. Set up a task with all the associated sub-tasks, and name it Onboarding. Whenever you need to onboard a new employee, find the task and use Jira Core's clone functionality to make a brand new task with all the associated sub-tasks. By renaming the task you'll know which new employee each task belongs to. You can even assign the sub-tasks to the relevant people, so that when you clone the task, Jira Core will make sure the relevant people are assigned to their tasks automatically.

Other customizations	You may decide that your onboarding should be more extensive, you may decide that instead of just getting the physical bits and pieces together for a new employee, you may want to also ensure that in their first week the get to have lunch with their team, sign off on all your statutory training, or get a building tour. You can customize your project's workflow to suit your needs, assigning the parent task to the person responsible for each step. You could decide to customize the fields available on the task, to make sure you record the information you need along the way (scores in training, asset numbers of any equipment that's been provided). You set up Jira Core to mimic how you do onboarding. It can be as complex or as basic as you like!

Got the hang of howJira Corecan be used? Let's take a look at another HR example, let's see how you could potentially set up a project to candidates that have applied for job openings you have.
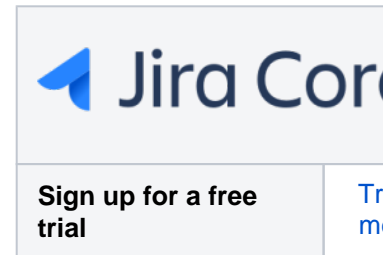
Use case	**Candidate tracking** - You advertise for a role that's become available, and you start receiving applications in the mail, through your website, and via email.
What Business project should I start with?	**Project Management** - This gives you a workflow of To do, In progress and Done, and accompanying issue types of Task and Sub-task.
How can I do this?	1. Create a task for each candidate applying for a role. 2. Add any accompanying CV's, resums or paperwork as an attachment to the task. 3. Depending on your hiring process, if the candidate is unsuccessful they may go directly to Done, or if you need to perform further tasks, like back ground checks, interviews, or wait for additional paperwork, the candidate may go to In progress. 4. Once you have a successful candidate, move them to Done and add a comment or label to say they've been successful.
How would this look?	
Can I make this easier?	Yes, you can! If you've integrated Jira Core with your email system, you can make sure that incoming emails regarding job applications are automatically created as tasks. Any accompanying attachments on the email will also be added as attachments to the task. Ever wondered why some job advertisements ask for your name, the position you're applying for and job reference in the email subject line? It's because this is exactly what they're doing, taking those emails and automatically converting them to "something" they can track and process! Why not use Jira Core to do it for you.

Other customizations	We know that processing a candidate for a role is never as easy as it sounds. Depending on the role, and any requirements you may need to do several interviews, background checks, make sure you've got the correct paperwork signed and returned... the list goes on! Jira Core allows you to customize the workflow to suit this, and you can assign the task to the relevant person at each stage. Want to use the same project for all your job applications, but worried after a while it'll get too large with too many tasks open? Well you can help sort your issues by assigning them defined labels for each open role. This will help you search the tasks easily and pick out the tasks that relate to the role you're hiring for. You could also add components to the project, which are like a grouping for your tasks. You can use components to automatically assign someone to each task that's placed within that component.

These are just some ideas of howJira Corecan help you with your process, and make it more visible and manageable.

Using Jira Core for Marketing projects

You work in Marketing, and your company is starting to win more contracts. There's a lot of work that needs done for each contract, and it can vary greatly! You're starting to lose visibility and need the ability to bring everything together and track the work that needs done, who's responsible, and when it's due by. You've heard of Jira Core and you decide to give it a whirl... but where to start?!

Jira Cor

| Sign up for a free trial | Tr m |

Let's take a look at what you could consider doing...

Use case	**Marketing campaign** - you need assets, locations booked, websites created, approval to list goes on!
What Business project should I start with?	**Project Management** - This gives you a workflow of To do, In progress and Done, and ac issue types of Task and Sub-task.
How can I do this?	1. Create a new task for each item of work that needs done. Give it a descriptive title to n everyone knows at a glance what it's for. 2. Add sub-tasks for any additional work that may need done, making sure you break big work into more manageable pieces. 3. When the sub-tasks are done, you can complete the parent task. And your work is don
How would this look?	
Can I make this easier?	Don't forget you can add comments to your tasks, and mention people in the comments by mention. This makes sure everyone is aware of the work and tasks that are needed. When and sub-tasks to track your work, it's also useful to remember you can view the Activity an page for your project by clicking on the project name in your sidebar. These views give yo of the issues in your project so you always know where you stand. If you need more detai can also create you own dashboard with all the statistics you need, and add a link to the d your sidebar.
Other customizations	Marketing tasks can be very diverse and very specific in their requirements. You can custo issues and make various fields 'required' to ensure they're always completed correctly. Yo fields in various ways (drop-down, multi-select, free text etc.) to make sure whoever create can only select from your pre-defined selections, or can complete the information as they s Notifications can also be customized so that the right people are always updated when the Make sure the tasks and project are set up they way **you** work.

Getting the hang of howJira Corecan be used? Talk to your administrator about more ways you can customize your project to make sure it suits how you work.

Using Jira Core for Operations projects

We all know operations can cover a large and varied amount of work, andJira Coreallows you to organize that work, no matter how simple or complex.Jira Coreworkflows can be customized to allow you to move through various stages on a non-linear path. Each step in the workflow is a status, and you move between statuses via a transition. Your transition could bypass several statuses to get you directly from To do to Done, or it could take you down a path that means once you select any other status other than Done, you have to complete more statuses to progress to Done.

Jira Cor

Sign up for a free trial
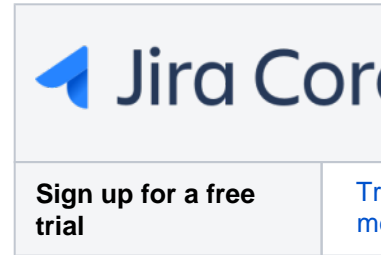
Sound a little complicated? Let's look at an example, say you run a small manufacturing business making bespoke metal parts. Some metal parts are simple, and all they require is one process, such as simple cutting with a laser, to complete the part, and some metal parts are more complex, requiring a series of operations to produce the finished part. How could you control these in one project?
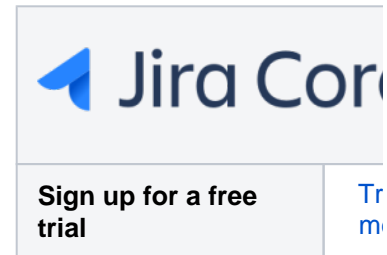
Use case	**Manufacturing** - metal fabrication
What Business project should I start with?	**Project Management** - This gives you a straightforward workflow of To do, In progress and Done, and accompanying issue types of Task and Sub-task.
How can I do this?	1. Create a task for each new piece of work that comes in. 2. Add a description that states what's needed in terms of raw material, and add any supporting drawings as attachments. 3. Create a sub-task for everything that needs to be done separately. You may need other parts cut and folded, or special parts ordered in. 4. Assign each sub-task to the relevant person responsible for making sure it's done. They can complete the sub-task when they've completed their task. Make sure you have these sub-tasks completed before advancing the main task. 5. Once all these sub-tasks have been completed, you can progress the parent task, and mark it as done once you have completed the work!
How would this look?	
Can I make this easier?	Well in this case, the flow is pretty straightforward. While work's in progress, the task is in progress, and when it's complete the work is done. Adding comments would be a great way keep track of the work, and where it is in the process. Assigning tasks and sub-tasks to the relevant parties helps make sure everyone knows when they have something to do!

Other customizations	Jira Core's workflows can be configured to suit your process, so in this case you may want to track all the work on one task. You could add statuses of all the main stages in metal fabrication, like Cut, Laser, Punch, Fold... right through to Paint, Inspection and then Done. If any stage isn't required, you can simply transition the task straight through to the next required stage. You could also customize the issue types, making a different sub-task for each stage, as some parts could probably progress through fabrication independently. You can even set up separate workflows for different issue types in the project! It's all about how **you** work, and how **your** team likes to do things. Jira Core just lets you do it and keep track of it.

Using Jira Core for Finance projects

Finance teams can useJira Coreto manage their processes and work, and making a few tweaks to a standard project template will make sure it suits the needs of you and your team. You could use a simple workflow when you need to generate invoices, or you could use a more complex process workflow for end of year financial reporting. MakeJira Coreyour one stop shop for all your financial process needs.
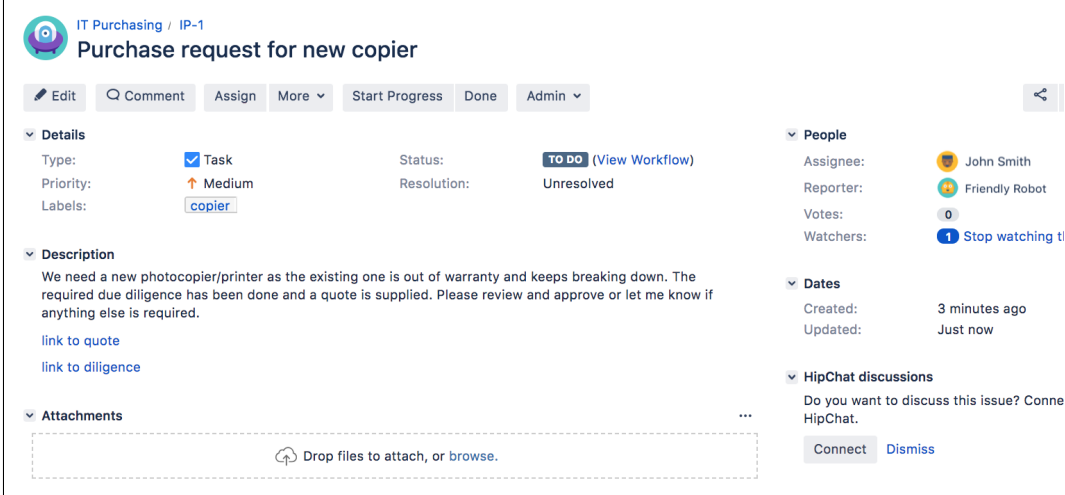
Jira Cor

Sign up for a free trial	Tr mi

Let's take a look at a simple invoice workflow...

Use case	**Invoice tracking** - make sure you know when to generate your invoices, and when to chase them.
What Business project should I start with?	**Project Management** - This gives you a workflow of To do, In progress and Done, and accompanying issue types of Task and Sub-task.
How can I do this?	1. Create a task for each invoice that needs to be generated. 2. Once you've generated the invoice and sent it, move the task to In progress, and add any details that need to be recorded to the comments of the task. 3. You can set a due date for when the invoice should be paid, or a date for when you want to chase it. 4. Once the invoice has been paid in full, you can move the task to Done!
How would this look?	
Can I make this easier?	If you need to make sure you record specific information for each invoice, such as the invoice number, add a field to the issue type Task called 'Invoice number'. This gives you a field you can search by when filled in, so it's easy to find the issue related to the invoice number you have. You can also add any attachments or emails to the issue to record any correspondence. You can add these along with any required comments. If you have one person who controls your invoicing, you can set them as the project lead to ensure they always get alerted when an issue is created.

| Other customizations	While you can add a field to the issue type to make sure you have a specific field to record your invoice number in, you can go one step further and make that field required when you want to move the task from To do to In progress. That means without a valid invoice number being issued, you can't move the task to In progress. You could do something similar to record when the payment has been made, you may decide you need to have a copy of the credit to your company account to close the task off as Done.

What about something a little bit more complex?Do you have a simple purchasing flow like Order item > Item received? Then you'll be happy with a simple workflow like Task management where a task is either To do or Done. But what if your process is more complex, like Item requested > Business case created > Business sent for review > Approval received > Purchase order created? UsingJira Core's Process Management project will give you an initial workflow to get started, and when your process matures, you can mature your workflow to make sure it always does what you need.

Use case	**Purchasing** - Creating and submitting a business case for approval
What Business project should I start with?	**Process Management** - This gives you a workflow of Open > In Progress > Under Review Approved > Done, and accompanying issue types of Task and Sub-task.
How can I do this?	1. Create a task for each new business case for approval. 2. Populate the task with all the relevant details, attaching any documents, spreadsheets additional information that you think will support the business case. 3. Moving the task to Under Review and assign it to your approver (this could be your ma or maybe a member of the purchasing department). 4. Once approved, the final stage could be to either raise a purchase request, or possibly transfer the data to another system to get the purchase completed. Either way, the issu Done and completed!
How would this look?	
Can I make this easier?	Yes! You can use a lot of Jira Core's functionality, especially comments and mentioning ot people (collaboration is alive and well!), to make sure you have all the information you nee complete the business case. Assigning the task to the relevant people at the right time as moves through the workflow makes sure it'll always appear on their to do lists, and if notific are set up, they'll even receive an email to let them know it's ready for them.
Other customizations	You may decide to change the workflow. You may need to add in a second approver. You also want to consider assigning the task automatically to the correct person as it progresse through the workflow. You may consider adding different issue types for each type of busin case that you accept, and assigning each issue type a specific, custom workflow (with the assignees per status of course!). Jira Core allows you to achieve this.

Think of anything else you may need to make sure you have everything recorded for your financials? Check out the Administering a project section of the docs for some ideas on other customizable areas of Jira Core.

Using Jira Core for Legal projects

Whatever your business requirements, if there's a process or a workflow involved,Jira Corecan help you get things in a more orderly state. Your organization may have a legal team, who are responsible for legal documentation, licensing, non-disclosure agreements and all the things required to make sure your organization complies with all required local, national and international regulations. Keeping track of all the requests, requirements, reviews and outstanding approvals can be quite tough, but by setting up aJira Coreproject and configuring it to how you work can help manage the work, and keep you on top of things.

Jira Cor

| **Sign up for a free trial** | Tr m |
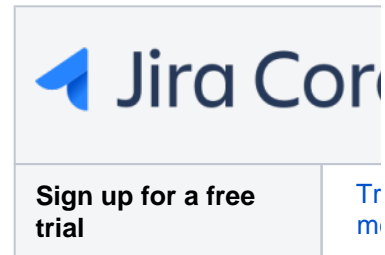
Let's take a look at non disclosure agreements, and how you could set up a project to keep track of the work you're doing.

Use case	**Documentation requests** - get your documentation requests in one place, to be actioned
What Business project should I start with?	**Project Management** - This gives you a workflow of To do, In progress and Done, and ac issue types of Task and Sub-task.
How can I do this?	1. Create a task for each document you need create. 2. Attach the document to the task. Remember to keep the document versioned in your w processing package! 3. Once the document is put together, get feedback in the comments section of the task. 4. When approved, transition the task to Done and finalize the document!
How would this look?	
Can I make this easier?	You may think there's something missing here, there's no review process? Well you can a ways. You can modify the workflow to add an 'In review' status, or you could add a resolut 'Done' status of 'Rejected'. That means that the person who prepared the document would transition the task back to In progress to work on it again.

Other customizations	Legal documents are very important, and need to be accurate and applicable to each case important to make sure the correct person reviews and approves the documentation. Your needs to make sure this happens. You could add a condition to the workflow transition tha specific person can transition it. That could be your approver. Remember, Jira Core allows your process. It can be as complex or as basic as you like!

Tips and tricks

Jira Coreis an extremely flexible workflow management system. You can customize and configureJira Coreto suit the needs of you and your team. If you haven't done so already, you should take a quick look at the Jira Core overviewpage to gain an understanding of whatJira Corecan do, and how it does it. I'm sure you have more questions on how to do certain things, how to configure elements ofJira Coreand what exactly is possible. A lot of the configuration and how to information is in theJira application administrationdocumentation. The content on this page is some tips and tricks, including links to more information, that should get you up and running and more productive faster!

Add a status to your workflow, like adding a "Review" status for documentation updates, is essentially changing the workflow. To do this you should read the Working with workflowsdocumentation, which will step you through adding a status, making sure you add transitions so that you can get to your new status, and give you more information on what you can do to that status to make it even more powerful. For instance, you may want to make sure that when you do move an issue to "Review", it's automatically assigned to a specific person. This can be achieved by adding a post function that allows you to update a specific field on an issue. Make the field the Assignee field, and select the specific person you want the issue assigned to. Bingo! The issue will now be assigned to the person when it's transitioned to that status. Take a look at Atlas sian Answers, it's full of useful information.

Jira Core has one Assignee field, and this allows you to assign an issue to one person at a time. This is intentional, as it means one person is responsible for that piece of work. It stops the cases occurring whereeither two people are working on the same thing, or neither work on it as they think the other person is working on it! That being said, there's a few ways to achieve the concept above. You could split the issue into sub-tasks, and assign each sub-task to a different person. Another slightly more complex option is to add another field to the issue called a "user picker" field, which allows you to select other users. The Assignee of the issue is still responsible for the work, but it does mean other people can be added to the issue and will show as working on it. See the tip on assigning issues to a group below for more information!

You can add additional fields to Jira Core's default issues, in fact you can even add additional issue types if you like. All this is covered in the Jira application administration documentation, just search for Adding a custom field. That'll walk you through which fields you can add to an issue, and how.

Issue field security is something that a lot of organizations request, but it's not something we support in Jira Core. You can read more about that decision on this issue

🔖 ~~JRASERVER-1330~~ - Provide field-level security permissions CLOSED .

You can set the security level on an issue however, visit the Jira application administration and search for Setting security on an issue for more information on that topic.

If you'd like to copy an issue, it's possible by using the Clone functionality. A cloned issue will create a link to the issue it's cloned from, and vice versa. You can read up on more details about issues and cloning them he re.

Dashboards! Jira Core comes with a variety of gadgets that you can configure to show your information how you need to see it. You can add these gadgets to your dashboard, and make that dashboard your homepage in Jira Core when you first log in. That way the first thing you'll see is the information that matters to you, and you'll know what you need to do.

Jira Core offers a powerful search function, and you can even save these searches as filters for use at a later stage. You can even share these filters with your team, so you're all looking at the same information. Keep on top of things together!

Yes there is! Jira Core comes with the ability to add a label to an issue, and labels can be used in searches. In fact, the label is 'clickable' which means that when you click the label, you'll be taken to a search that shows all issues with that label. You can then refine that search if needed.

A word of caution though, check with your project administrator as to your organization's policy on using labels. There may be a policy in place regarding how this field should be used, as some organizations use it as part of a process.

Jira Core is designed so that issues must beassigned to a single individualto prevent tasks from being overlooked. A team lead or manager should assign issues out to individuals, or your users will pick from a list of issues that they have the option to take on.

However, if you want to configure Jira Core to allow issues to be assigned to multiple users there are a few option for doing so:

- Managing Issues via a Queue
- Managing Issues via Group Ownership
- Managing Issues via a User Account
- Managing Issue via Sub-Tasks

It is easy to still setup a queue the a group can pick from, or affiliate an issue with group in addition to having it assigned to an individual within that group:

Managing Issues via a Queue

You can configure your Jira Core project to assign issues to an 'Unassigned' "queue" by default, which your users can then pick issues from.

To do this, set up the following:

1. Configure your Jira Core project to allow the 'default assignee' to be 'Unassigned'.
2. Ensure that 'Allow unassigned issues' is set to ON in yourGeneral Configuration settings(**Administrati on > Global Settings > General Configuration**).
3. Set any issues that you want to be in the queue to be 'Unassigned'.
4. Create a dashboard page with a filter that lists all 'Unassigned' issues, share the dashboard page and request that interested members of the group display the shared page on their dashboards.

Managing Issues via Group Ownership

You can add a custom field to store which users and groups should be associated with a given issue. This is particularly useful for projects where a team owns all issues of a particular type.

To do this, set up the following:

1. Add agroup picker custom fieldto your issues.
2. Configure an email notification in your project'snotification schemeto be sent to the 'Group Custom Field Value'.

An issue can now be "assigned" to the group by selecting the appropriate group in the group picker. An email notification will be sent to the group.

🛈 *Another option is to add auser picker custom fieldrather than a group picker, and assign multiple users to an issue. However, you will then have both the Jira Core default user field and custom user field for your assignees.*

Managing Issues via a User Account

You can create a Jira Core user account to represent a group of people (e.g. 'developers') and assign issues to this user.

To do this, set up the following:

1. Create a Jira Core user to represent the group.
2. *(Optional)*Create an email mailing list for this group (not a Jira Core function) and set the mailing list email as the Jira Core user's email address.
3. Create a dashboard page showing issues assigned to this user, share the dashboard page and request that interested members of the group display the shared page on their dashboards.

An issue can now be assigned the new "user" representing the group and your users can track the issues on their dashboards. If you have set up a mailing list, your users will also be notified by email.

Managing Issue via Sub-Tasks

If you have a task managed by different users then you are able to break the combined task intoindividual subtaskswith their own single assignees.

Yes it is. Jira Core comes with two methods of grouping issues, Versions and Components. You can read more on these in the Administering a projectsection of the documentation. Essentially versions and components allow you to group issues, and the main distinction is that with a component you can assign a user as a default assignee when the component is added to the issue, and versions can have a start and end date.

How do I build the workflow I want?

So you've got to the stage where you need a little more from your workflows than just "To Do", "In Progress" and "Done"? Great! Jira Core lets you customize and configure the workflow of your project so that you can move work through it the way you need it to. You need be a JIRA administrator to change a workflow, and we have all the information you need on changing a workflow in the Jira administrator documentation, but what you probably want to know is what is possible with my workflow? What exactly can I do? Am I confined to just adding steps? We'll take you through some steps to creating a great workflow, and give you some information on what's possible.

On this page:

- What's in a workflow?
- How do I create the workflow I need?
- Make every transition count
- Building your workflow
- Tips and tricks

What's in a workflow?

First, lets take a look at what defines a workflow. A workflow has four unique components: statuses, transitions, assignees, and resolutions. As an example, libraries have well-known workflows surrounding materials they lend out to the public. Each item could be stored in JiraCore as an issue, and administered in the following simple workflow:
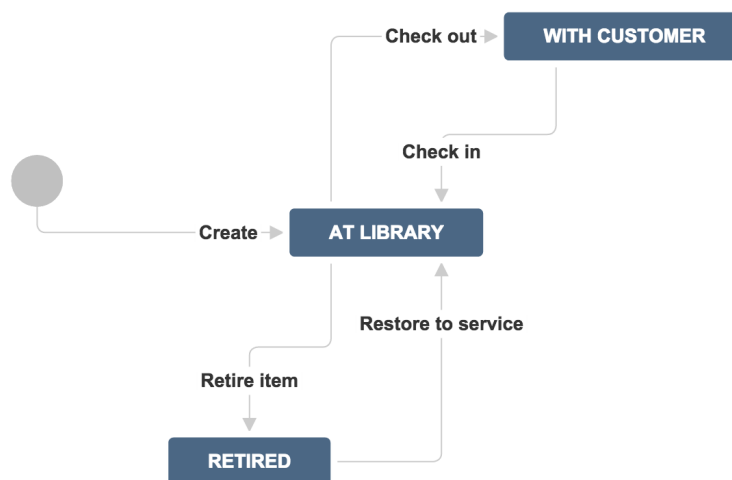
Statuses: where

Statuses represent the position of the issue within a workflow. In the library example, a book can be in one of three places: at the library, with a customer, or retired from inventory. Every status should be a unique state in your workflow.

Transitions: how

Transitions are the bridges between statuses; the way a particular issue moves from status to status. At the library, books are loaned out by librarians, returned by customers, and evaluated by library staff to see if theyre fit for circulation. Note that customers cannot evaluate if an item is fit for inventory that must be decided by the library staff. With JIRA Core, you're able to set workflow permissions that allow only the right people to transition an issue.

Assignees: who

Workflows guide how people work together. In JiraCore the assignee dictates the responsible party for any given issue. In the library example, we would set the assignee to the borrower every time a book is loaned out, so that the library knows which customer has that book. When the book is returned from inventory, we would then remove the assignee when we transition the book back to inventory.

Resolutions: why

Resolutions explain why an issue transitions from an open status to a closed status. The act of retiring a book removes it from inventory, therefore, we can put a resolution on that book. In this example, we might use resolutions such as *damaged*, *out of date* or *lost*. The two biggest mistakes new JiraCore administrators make are:

- **Confusing resolution with status** Status describes where an item is in the workflow whereas resolution explains why an item is no longer in flight. In order to effectively use these features to retire a book from circulation, our librarian should create a **status** called *retired*, and options for **resolution** that include things like *lost*, *damaged*, or *out of date.* Searching for everything thats retired (or resolved) with the ability to search for *why* its retired via the resolution options gives much better reporting metrics. Thats why JIRA Software uses **resolution** to know if the issue is considered inactive by the organization. The best way to check to see if youve got it right is to use the created versus resolved gadget. You can have a workflow between when an issue is tagged with a resolution and officially closed. Many organizations verify resolved issues to ensure they're resolved for the right reasons.

- **Setting resolution correctly** You must set a resolution when an issue moves from an open state to a closed state. Likewise, resolution needs to be removed when an issue gets reopened. At the library, if the librarians decide to put the book back into circulation, they would need to remove the resolution from that book. For example, a book that is able to be checked out should not have a resolution of *lost*.

How do I create the workflow I need?

Building the workflow you need can be challenging. Remember, it's all about building what you and your team need to work more efficiently and effectively. That may involve you and one other person, or it could involve several departments and many people. Here's a few pointers to get you going.

Gather all of your stakeholders

Workflow is about scaling culture, and culture is about people. Whenever it comes time to build a process around a set of people, identify all of the stakeholders in that workflow. For example, if youre trying to build a workflow between product management, software development, and support, ensure you have one representative from each organization in the meeting. As the person designing the workflow, spend time talking with each stakeholder about whats important to them. Before the meeting starts, draw a draft workflow on the whiteboard, then walk through each case in the meeting and gather stakeholder feedback. Workflows are touchy as they govern how people work, so be patient, and your investment will pay off later.

Use a whiteboard in low fidelity so its easy to get feedback and make changes at this stage.

Keep the workflow simple: less is more

Most of the time, stakeholders want to have statuses for each part of the workflow. Generally, thats a good thing, but remember: each status adds more transitions and complexity to the workflow. Workflow is designed to make things simple and scalable. Whenever you're adding a new status to a workflow, ensure that the stakeholder has no other option but to grow the workflow. Lets look at two examples.

- For many news papers, articles needs to be reviewed and this is an important part of the editorial process. Jane, the editorial manager, wants to add a specific status called *article review* so that its clear to the team what issues are still being written, and which issues are waiting review. Reviewing articles is distinctly different from writing articles. It makes sense to add a new state as the review state will indicate an article has been written, and a different person (assignee) is now responsible for review.
- Bill, the editor, wants to add a new status called *rejected* for all issues that dont pass review by his team. Id advise against doing this, as the writing team can simply reopen any issue that fails review. An alternative option is to move the issue into a *resolved* state with the resolution *rejected*.

As a JiraCore administrator you owe it to yourself and your users to keep workflow simple.

Make every transition count

JiraCore has a number of options administrators can employ when transitioning issues between states.

- **Conditions** Conditions control who can perform a transition. For example, only someone in the library can retire a book from inventory.
- **Validators** Validators ensure that a transition can happen given the state of the issue. For example, if a book is to be retired because its allegedly damaged by a customer, wed like to ensure the book has been checked out at least once to verify that claim.
- **Post functions** Post functions execute actions on issues after both conditions and validators pass. The most common post function is to clear the resolution when reopening an issue. In the library example, we would use a post function to clear the reason a book was retired when it got put back into service. JiraCore keeps a detailed history of issues, so we could look up when the issue was retired and why.
- **Assignees** Whenever a transition occurs, always ask who the new owner of the issue is. Ensure that in every transition there's a default action to route the issue to the right place. Users can override the default action as well.
- **Properties**JiraCore recognizes somepropertieson transitions. The most common one is to limit resolutions displayed to the user on a given transition. For example, we might want the resolution*scratc hed*to show up for disk media when retiring, but not for books.

Building your workflow

Check out the Jiraadministrator documentation on workflows for all the information on how to build the workflow you want. Once youve built your workflow in JiraCore, you should take the time to ensure the workflow is robust and doesn't have any dead ends, or unexpected behaviors. You'll want to share the workflow with all of your stakeholders for a final round of feedback. You should also enable the option in JiraCore to easily show where the current issue is in the workflow. All users have to do is click*view workflow*.

Tips and tricks

There's a few things you should check, and a few tricks you can use you get the most out of your workflows:

- If you intend to change the workflow you're using for your project, make sure that the workflow isn't being used by any other projects! Any changes made to a shared workflow will impact all the projects using it.
- If you already have a workflow that works for you, and you want to use it in another project, you can either create the new project and select "Create with shared configuration", or once you've created the project you can change the default workflow to the existing workflow that you want.
- Once you edit your workflow, remember to publish it! Publishing the workflow effectively makes it active on your project.

Installing Jira Core

Jira Corecan be customized to suit your needs, and can also be extended and linked with other applications to provide you with the perfect solution to track all your work in one place.You can install Jira Core Server anytime by downloading a trial version, selecting the server installation type that suits your needs, and choosing the number of users that you want the installation for.

Once you have downloaded the installer, follow our Installation guide to get up and running.

<div align="center">Download</div>

If you experience any trouble with your installation or you have any question, please contact Support .

Jira applications overview

Jira application overview

The Jira family of applications are built on the Jira platform.Jira Coreis the default application of the Jira platform, and will always be present in a Jira instance. You may also choose to include other applications in your instance, such asJira SoftwareandJira Service Management. A user may require access to one, all, or any combination of these applications.

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	• Available to all licensed users
Jira Software	Software projects	• Integration with development tools • Agile boards • Release hub for software versions
Jira Service Management	Service projects	• 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 asBitbucket and FishEye, as well as agile boards, but this information is only viewable by a Jira Softwareuser. A Jira Coreuser 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 Softwareuser would not be able to see any Jira Service Managementapplication-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 toJira Coreusers. 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 areJira Software-specific features. To know how to allow anonymous users access to projects, seeAllowing 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
Business Projects	Project level	Create	✓	✓	✓	✗
		View	✓	✓	✓	✗
	Issue level	Create	✓	✓	✓	✗
		View	✓	✓	✓	✗
		Comment	✓	✓	✓	✗
		Transition	✓	✓	✓	✗
	Jira Gadgets	View	✓	✓	✓	✗
Software Projects	Project level	Create	✗	✓	✗	✗
		View	✓	✓	✓	✗
	Issue level	Create	✓	✓	✓	✗
		View	✓	✓	✓	✗
		Comment	✓	✓	✓	✗
		Transition	✓	✓	✓	✗
		View Development Information	✗	✓	✗	✗
		View Release information	✗	✓	✗	✗
	Board level	Create	✗	✓	✗	✗
		View	✗	✓	✗	✗
	Jira Software gadgets	View	✗	✓	✗	✗
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 users.

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:
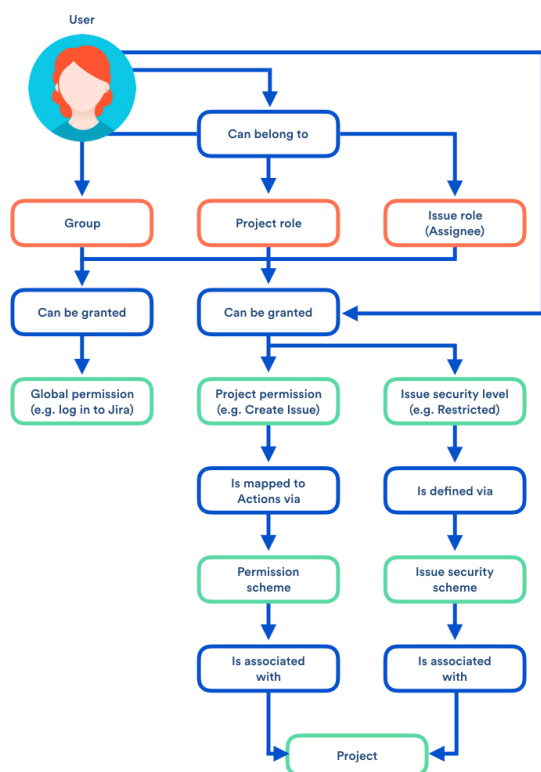
Who can set permissions?

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

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.
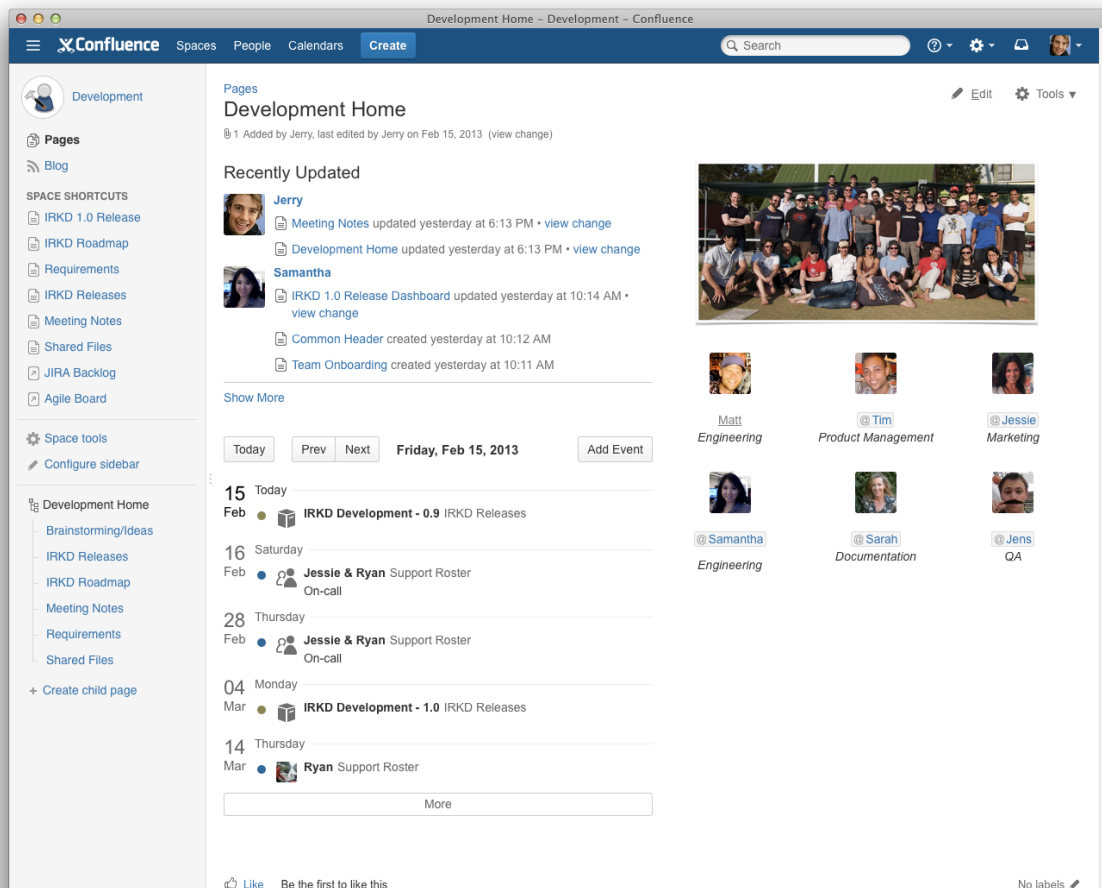
Why use Confluence with Jira?

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. Th Requirements Blueprint helps development teams create, discuss, and organize their product require 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

39

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.

Using Jira applications with Advanced Roadmaps for Jira

Advanced Roadmaps forJira(formerly Portfolio) provides a single, accurate view for planing and managing initiatives across multiple teams and projects with ease. You can:

- Plan top-level business initiatives and break them down into lower level deliverables for development teams
- Track investments across strategic themes to ensure that those investments align with business priorities
- Generate realistic delivery forecasts with automatic scheduling algorithms
- View the progress of any initiative based on real-time, accurate data fromJira
- Drive accurate and realistic capacity planning by defining teams and allocating work based on skills and availability
- Scope releases in a matter of clicks
- Make fast prioritization and tradeoff decisions to instantly see the impact of plan changes
- Minimize productivity loss by easily reacting to the ever-changing needs of your business in real time
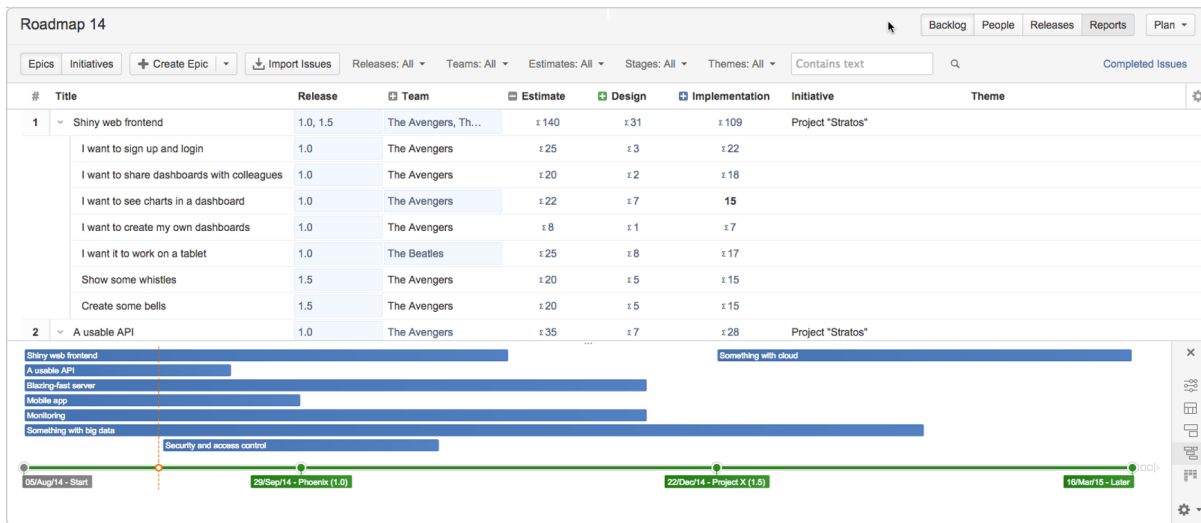

Plan automatically

- Set priorities, estimates, and target dates to instantly see when you can ship releases based on your commitments
- Automatically optimize your plan and get suggestions on ideal resource allocation to create a realistic forecast
- Account for dependencies, resources, parallel vs. sequential activities and the realistic number of people that can work on a single item to create an optimized roadmap
- Use themes to categorize your backlog items by strategic focus areas, value streams, or investment categories to see relative resource allocation between themes


Avoid bottlenecks

- Identify and avoid bottlenecks and potential holdups by accounting for dependencies across teams and projects
- Model skills and define who can do which type of work to avoid unrealistic resource loads
- Automatically account for team memberavailability, including time off, training and inter-team commitments
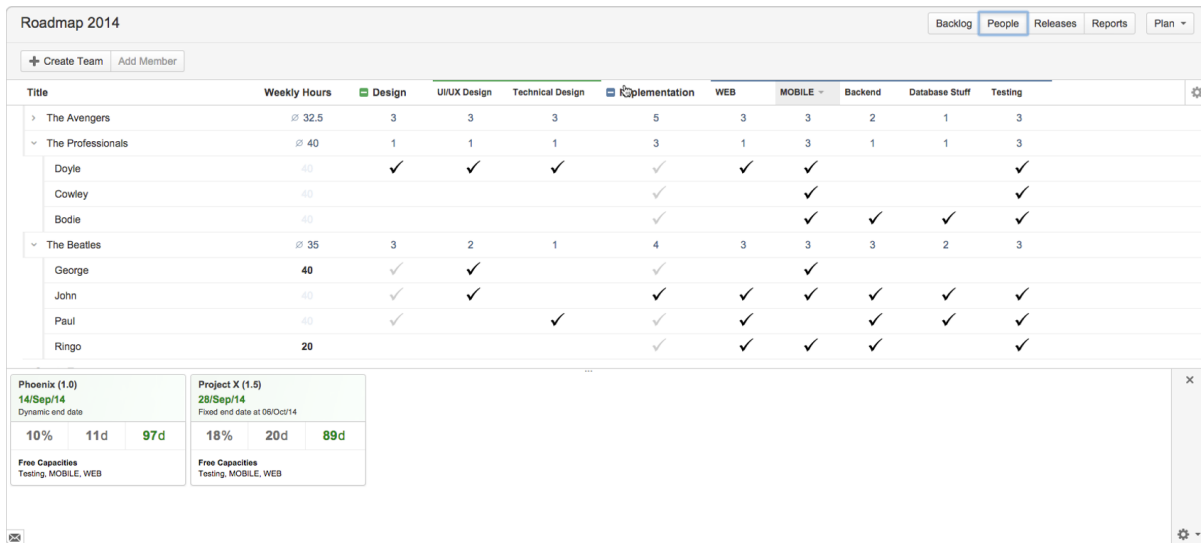
Keep up to date

- Keep the long term plan in sync with reality by adjustingdelivery dates, team member resources and dependencies using up to date data
- Changes are displayed in real-time across all projects, giving you a comprehensive view of your entire roadmap
- Having an up-to-date schedule gives your team a transparent understanding of what's next, lending clarity to your decision-making
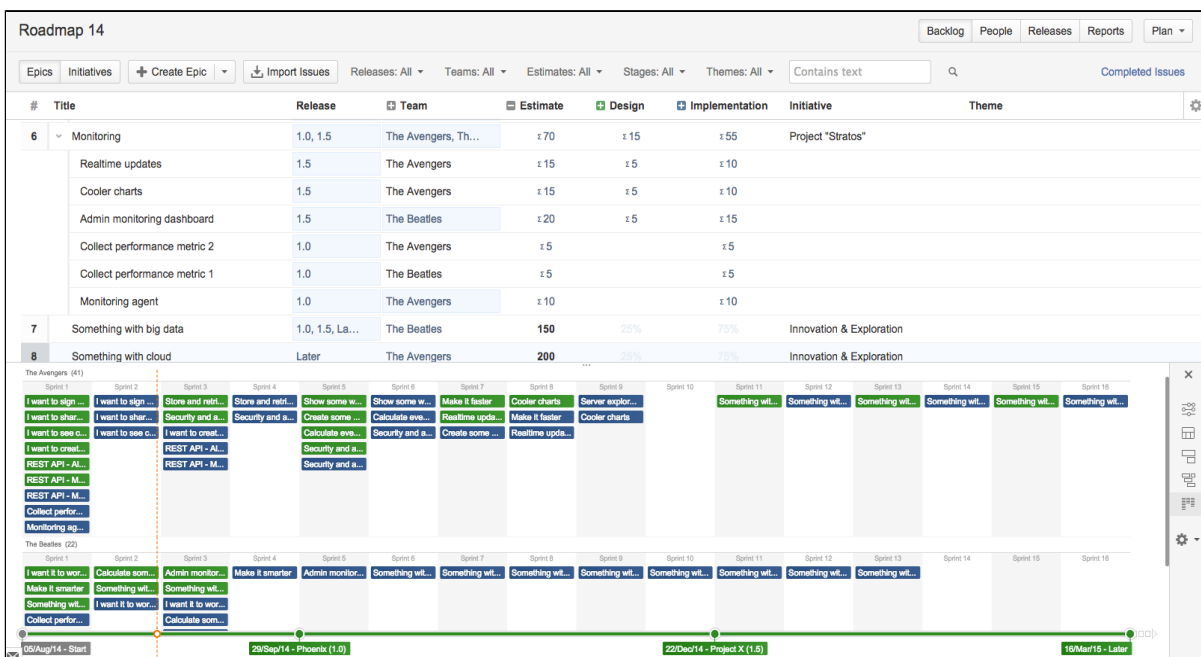
Make realistic commitments

- Confidently make commitments for scope and ship dates by using your more reliable forecasts
- Adjust dependencies to quickly check the impact across all of your projects, for example, if a critical feature is taking longer than expected
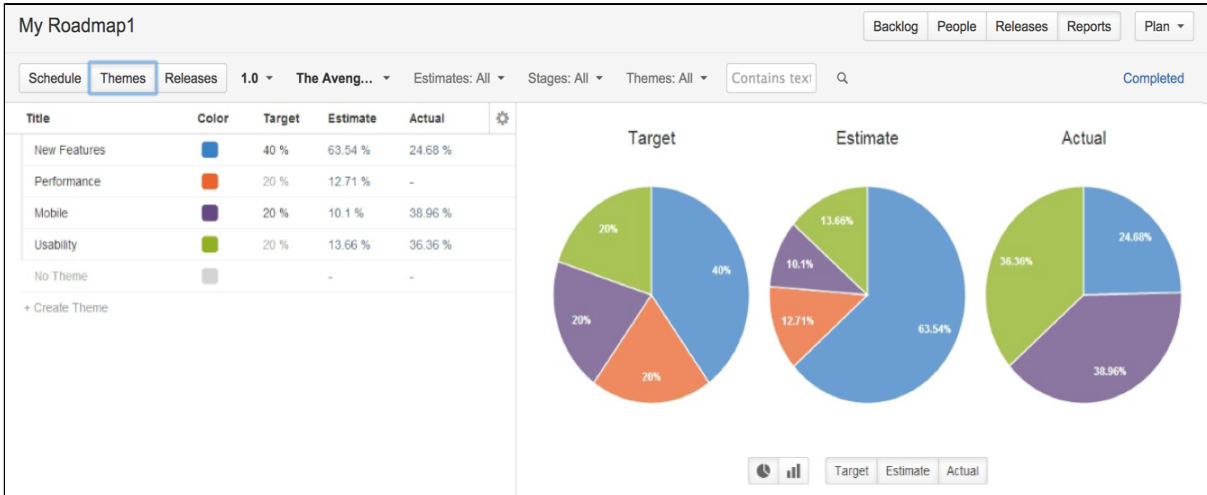
Model changes

- Quickly visualize what different scenarios and decisions mean to your projects
- Evaluate new change requests by seeing the impact on scope, dates, resources and cost commitments
- Model different scenarios without affectingthe underlying data in your Jiraprojects

You can find more information about managing your portfolio here:Advanced Roadmaps Home

Getting started with Jira Core

Welcome to theJira CoreGetting started tutorials! We know you want to get up and running withJira Coreas quickly as possible, so we've designed our Getting started tutorials to take you through the basic concepts and tasks you'll need to know inJira Core.

This is a hands-on tutorial, and we'll be taking you through the basics ofJira Core, and its three main user roles.

- If you're evaluatingJira Core, we suggest you do all three tutorials.
- If you've been invited toJira Core, we suggest you do the tutorial that best suits your user role.

So what isJira Core?

Jira Coreis a workflow management system which allows you to set up unique processes that suit the way you work. At the heart of all systems are workflows, moving packets of work from A to B.Jira Coreallows you to make your workflow as easy or as complex as you need, giving you the freedom to concentrate on the work, not the process.Jira Corecan be used in a variety of ways. For example, you could use it to run a t-shirt business, setting up workflows that can manage your internal processes such as your design process, your sales process, your manufacturing process, and even controlling your stock. The beauty ofJira Coreis that the only constraints on your workflow are your processes!

What types ofJira Coreusers are there?

Jira Coreusers can be grouped into three types:

Administrators	Project administrators	Users
Administrators are responsible for the configuration of Jira Core. They control the initial set-up, the look and feel, and the configuration of project workflows, issues, and generally what the end user will see and be able to interact with. Get started as an administrator	Project administrators are responsible for configuring their projects. They can administer projects, change the look and feel, and make various configuration changes to the project. A project administrator can't create a project unless they're explicitly given this permission by an administrator. Get started as a project administrator	Users are responsible for working in specific Jira Core projects. Users are given access to a project's issues, and, depending on their permissions, work on the issue by commenting on it, transitioning it through its workflow, and closing it when complete. Get started as a user

Getting started as an administrator

Jira Coreadministrators should complete this tutorial to understand the basic concepts of managingJira Core.

Here's what you can count on learning:

1. How to set up an evaluation instance of Jira Core
2. How to add new users
3. How to create a project and customize it
4. How to manage permissions

By the end of this tutorial, you'll have a fully functioningJira Coreinstance, several users with different access permissions, and you'll have created your first project.

| **Audience** |
| Administrators |
| **Time** |
| 45 minutes |

Ready to dive in and get your hands dirty? Get started and learn how to set up your ownJira Core.

Let's get started

Setting up your instance

1. Setting up your instance
2. Creating a project
3. Adding new users
4. Managing permissions

As a first step, you'll set up an evaluation instance ofJira Core. We'll guide you through three simple steps to get JIRA up and running in no time!

If you're ready to set up a production Jirainstance or you want more control, check out ourfull installation guides.

Before you begin

Our installers come with all the bits and pieces you need to run the application, but there's a few thingsyou'll need to get up and running:

- A computer or laptop with a supported operating system - you'll be installing Jiraso you'll need admin rights.

    You can install Jira on a Windows or Linux operating system.

    Apple Mac isn't supported for production sites, but if you're comfortable setting up applications on your Mac from scratch, you can download the `tar.gz` file and follow the instructions forInstalling Jira applications on Linux from Archive File as the process is similar.

- A supported web browser - you'll need this to access Jira, we support the latest versions of Chrome and Mozilla Firefox, Internet Explorer 11, and Microsoft Edge.
- A valid email address - you'll need this to generate your evaluation license and create an account.

Ready to get going? Let's start with grabbing the installer.

Download the installer

Head towww.atlassian.com/software/jira/core/downloadand download the installer for your operating system.

Install your Jiraapplication

The installer allows you to choose Express or Custom installations.

The Custom installation allows you to pick some specific options for Jira, but for this guide we'll use the Express installation.

1. Run the installer - we recommend running with a Windows administrator account.
   If prompted, make sure you allow the installer to make changes to your computer. This will allow you to install Jira as a service.
2. Choose **Express Install**, then click **Next**.
3. Once installation is complete, it will ask you if you want to open JIRA in your browser. Make sure this option is selected then click **Done**.
4. Jira will open in your default browser, and you're ready to start the set up wizard.

1. Change to the directory where you downloaded Jira then execute this command to make it executable:

```
$ chmod a+x atlassian-jira-software-X.X.X-x64.bin
```

Where `jira-software.X.X.X` is is the Jira version you downloaded.
2. Run the installer - we recommend using `sudo` to run the installer as this will create a dedicated account to run Jira and allow you to run Jira as a service.

```
$ sudo ./atlassian-jira-software-X.X.X-x64.bin
```

3. When prompted, choose **Express Install** (option 1).
4. Once installation is complete head to http://localhost:8080 in your browser to begin the setup process.

Set up your Jira application

The set up wizard is the last step in getting Jira up and running. You'll need your email address to generate your evaluation license.

1. Select **Set it up for me** and then **Continue to MyAtlassian**.
   This will allow Jira to set up everything it needs to run, including an H2 database.
2. Create an account (or log in if you already have an Atlassian ID account).
3. Follow the prompts to **generate a license** for the Jira application you want to try, and apply it to your new installation.
4. Enter and confirm the details you want to use for your administrator account, and click **Next**.
5. It will take a few minutes to get everything connected and operational.

That's it! Let's now create your first project.

Next

Creating a project

AJira Coreproject is a container that holds issues. Issues can be viewed as the packets of work required within a project. To create issues, you must have an available project to contain them. Jira Corecomes with several default project types with preconfigured workflows and issue types, so you can quickly get your project up and running.In this step of the tutorial, you will use the project management template to help your team plan, organize, and collaborate on their work.

Note that creating and configuring a project is done by an administrator. A project administrator controls user access to the project, and can only configure certain aspects of the look and feel of the project.You should still be logged in toJira Coreas an administrator from the previous step. If not, log into your administrator account.

Create a project

When creating a project, you will need to give it a name, a key, and a project lead. The title can be as descriptive as you want, and the key should be something meaningful. The project lead is usually the project manager, but can effectively be any user you select when creating the project.

1. If you're following from the previous step, you should see different project options on the welcome screen.Click**Create new project**.
2. Select **Project management** as the project type.
3. Enter**Dragon Design Tees**as the project name. Note thatJira Corecreates a Project key for you, but you can overwrite this if you want to.
4. Select**Submit**to create your new project.

> **About project keys**
>
> Each project has a unique *name*(e.g. **Dragon Design Tees**) and a unique*key*(e.g. **DDT**). The project key becomes the first part of that project's*issue keys,*e.g.**DDT-1, DDT-2,**etc.

Customize your project

In this step, you will be customizing your project avatar and project details to help your team identify the project more easily. These customizations are helpful if you have several projects in yourJira Coreinstance.If you have navigated away from your project, simply go to**Projects**>**Dragon Design Tees**.

1. Click**Administration** (⚙)**> Projects**.
2. Select**Edit**next to your project.
3. Click the**Avatar**image.
4. Select an available icon or upload an image.
5. Enter a URL and Description for your project to make it easier for your team to identify. Note that these fields are optional and only for display.
6. Click**Save details**to save your changes.

Congratulations! You've now created and customized your first project. Next, we'll add users to your project and look at how you can set up and restrict access to projects.

Next

Adding new users

1. [Setting up your instance](#)
2. [Creating a project](#)
3. Adding new users
4. Managing permissions

Working alone isn't much fun, so let's add some test users to yourJira Coresite. You can add users directly, or allow new users to sign up themselves.In this step in the tutorial, you'll add three users directly to your site.

Add a few users

You will be adding three users:**Jason, Kate**and**Emma**. You can add more or choose your own usernames if you like, but please note that we will be referring to these usernames later in the tutorials. You can always disable or delete any users you set up.

If you've logged out of Jira Core, log in with the administrator account you created.

1. Navigate to the **User Management** screen by selecting**Administration (⚙)> User management**.
2. Choose**Create User**to add a new user. Specify the username as**jason**. Set the rest of the fields to whatever you want. You're going to be creating a couple more users, so check the**Create another**che ck box before selecting**Create user**.
3. Now create two more users, with the usernames**emma**and**kate**, following the same process outlined above.

You should have a screen that looks something like this:

Users						Invite users	Create user

Filter users	In group	Application access	Status	Users per page			
Name, username or emai 🔍	Any ▾	All Users ⇕	All Users ⇕	20 ⇕	Filter	Reset	

Displaying users **1** to **4** of **4**.

Full name	Username	Login details	Group name	Applications	Directory	Actions	
Emma	emma emma@atlassian.com	Never logged in	jira-core-users	JIRA Core	JIRA Internal Directory	Edit	⋯
Jason	jason jason@atlassian.com	Never logged in	jira-core-users	JIRA Core	JIRA Internal Directory	Edit	⋯
Kate	kate kate@atlassian.com	Never logged in	jira-core-users	JIRA Core	JIRA Internal Directory	Edit	⋯
Tomek	admin tomek@atlassian.com	**Count:** 25 **Last:** 3 minutes ago	jira-administrators jira-core-users jira-servicedesk-users jira-software-users	JIRA Service Desk JIRA Software	JIRA Internal Directory	Edit	⋯

> ⓘ Note that usernames are **not** case sensitive. Emma can enter her username as Emma, emma, or even EmMA to log into Jira Core. Passwords, on the other hand, are case sensitive.

Well done! You've added three new users to your Jira Coreinstance. Next, you'll learn how to manage access to your project with site and project permissions.

[Next](#)

Managing permissions

You won't want every user in your team to have the same level of access toJira Core. For example, you may want to restrict who can administerJira Core, or prevent users from viewing a project. In this step, you will learn about the different permissions inJira Coreand set permissions for a new project.

Overview of roles, groups, and users

A role is a project-specific set of groups and/or individual users. In our example of the design project in the t-shirt business, all product managers need to be able to assign work (issues) across all projects, while senior designers need to be able to assign work on specific design projects.InJira Core, you can define a product manager role that includes all product managers. You can then define a set of permissions with the 'Assign issue' permissionfor this role, and apply this set of permissions to all projects. Individual senior designers can be added to the product manager role on each project, as needed.

Overview of global and project permissions

Global permissionscover a small set of functions that affect all projects inJira Core(for example, permission to administerJira Core). They can only be assigned to groups:
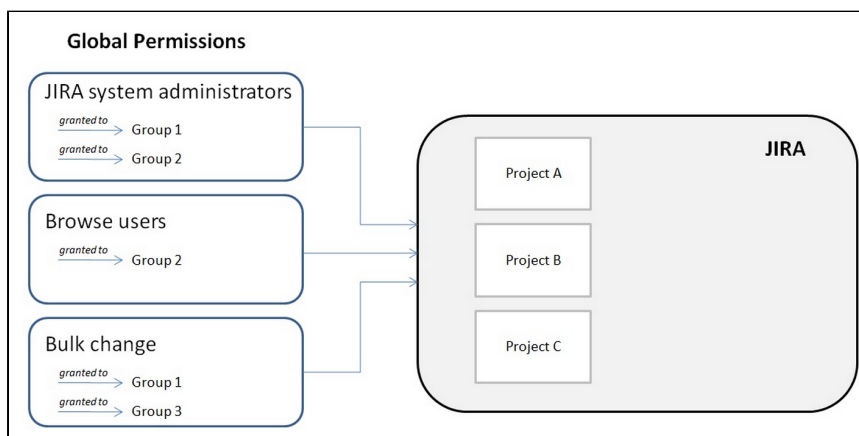
Projectpermissionscover a set of more granular functions that affect a single project inJira Core. For example, permission to create issues in a project. They can be assigned to groups, users and project-specific roles:
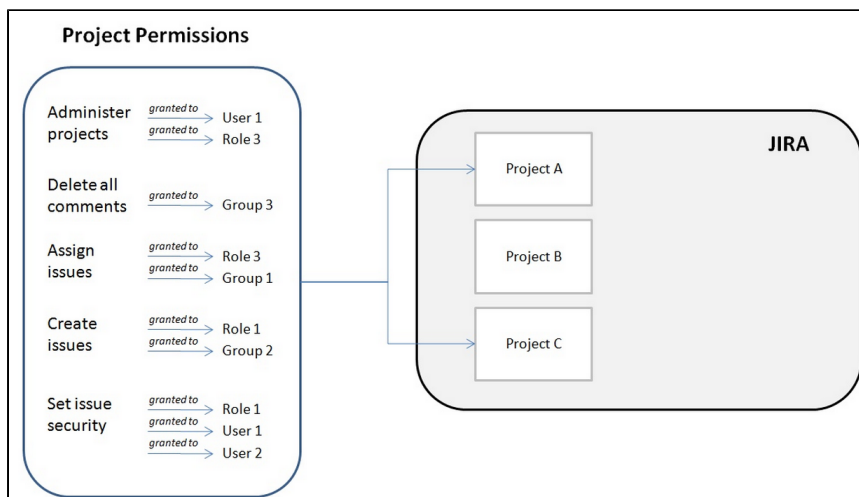
Now let's put this into practice! You're going to go through the tasks involved to use project permissions to hide a new, secret t-shirt design project from some of your users.

Create a new project role

This project role will only contain users that you want to view a particular project. We will assign permissions to this role in the next step.

1. Select **Administration** (⚙)> **System**.
2. In the security section, select**Project**roles.
3. Below the existing project roles, add another project role named "Review". Leave the Description field blank for now and select**Add project role**.
4. Select **Manage Default Members** and then, under Default Users, select **Edit** to add yourself and**Jason** to the Review project role.Do not add Kate or Emma.

Configure a new permission scheme

The 'Browse Projects' permission controls whether a user can browse a project, i.e. whether they can view the project. Let's assign this permission to your new project role.

1. Select**Administration** (⚙)> **Issues** > **Permission Schemes**.
2. Copy the**Default Permission Scheme**.
3. Edit the copied permission scheme and change the name to**Confidential Permission Scheme**. Select **Update**.
4. Select**Permissions**for the**Confidential Permission Scheme**. For the**Browse Projects**permission:
   - Click**Remove**for 'Application Role (Any logged in user)'.
   - Click**Edit**, select **Project Role,**and choose **Review**in the drop-down. Click**Grant**.

Associate the scheme with a new project

For the last step, let's associate the permission scheme with your new project.

1. Select**Projects**>**Create Project**and choose **Task management**.
2. Name the project**Top Secret Tee**and**Submit**.
3. In the bottom-left corner, select**Project settings > Permissions**.
4. On the Default Permission Scheme screen, select**Actions**>**Use a different scheme**.
5. Set the**Scheme**to**Confidential Permission Scheme**,and click**Associate**.

The only users that will be able to browse your new project are Jason and yourself. Note that default members are only added to a role for new projects. You can also use this approach to restrict users from creating issues, adding comments, closing issues, etc, in a project.

Well done! You created a project permissions scheme and applied it to a project.

You've now completed the Administrator Getting started tutorial. We suggest you complete the Project Administrator tutorial as well, so you have a better understanding of how your team will be using Jira Core. So put on your project administrator hat and let's get started!

I'm a project administrator

Getting started as a project administrator

Jira Coreproject administrators manage specific business projects that they have been assigned to in Jira Core.Project administrators can customize their projects, add and remove users, and perform certain configuration tasks, like adding versions or editing their workflow.

This tutorial will teach you some basics on how to manage your project and navigate around it's project settings. Here's what you can count on learning:

1. How to customize your project
2. How to add users to your project

Ready to exercise those product administrator muscles? Get started by customizing your project.

[Let's get started](#)
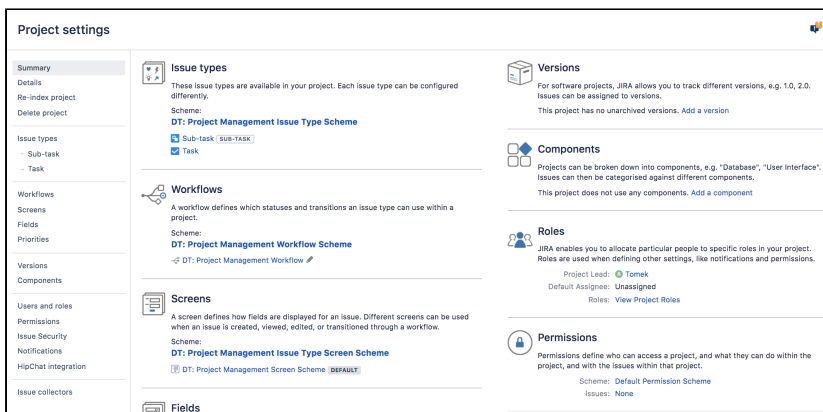
Customizing your project

1. Customizing your project
2. Adding users to your project

As a project administrator, you have the ability to edit project role memberships, project versions and certain project details (project name, project description, project lead and URL). In this step, you'll customize your project by changing the name, the URL and the avatar.

If you're continuing from the Getting started as an administrator tutorial, you'll already have a project called Dragon Design Tees. For this tutorial, use the Dragon Design Tees project or another project to which you have been given project administrator access.

Access your project's administration page

1. Select **Projects** > **Dragon Design Tees** (or the project you wish to administer).

2. Select **Project settings** (⚙) to open the Project Administration page:

> **Accessing your projects**
>
> You can also access your projects by selecting ⚙ > **Projects**. You will see a list of the projects you administer, and can select one you want to work on.

Customize your project

Customizing your project will make it stand out, and make it easier for your teams to recognize if they're working on multiple projects.

1. On the Project Administration > Summary page, select **Edit project**.
2. Click **Select image** to change your project avatar.
3. Select an image from those displayed, or upload an image specific to your team.
4. Note that you can also change the project name and description in this Edit Project dialog.
5. Select **Update** when you have made all changes.

Your changes will appear to all users who can access the project.

Success! You've customized your project avatar. Now, let's add a user to your project.

Next

Adding users to your project

1. Customizing your project
2. Adding users to your project

As a project administrator you can add groups or individual users to various roles in your project. Roles are project specific, so adding a group or individual user will only affect your project. You can also use the same process to remove access from groups or users. It's important to note that you can only add**existing**groups and users, you need to be aJira Coreadministrator to create the groups or users.

Viewing and changing project access

In this step, you will add one of your users (Emma) to the Administrators role, so she can help manage your project.

1. In your project, select **Project administration** > **Users and roles**.
2. Select Add users to role.
3. Search for Emma. You can add multiple users and groups, and delete those you have accidentally added in this Add users dialog.
4. Select the Administrators role and select **Add**.

Emma will now have administrator access to your project. You can use the same process to add users or groups to the Administrators role.If you would like to give users restricted access to your project (so they can only view issues they're assigned to, for example), you can ask your administrator to create a new project role.

Well done! You have added Emma as an administrator on your project.

You've now completed the Project Administrator Getting Started guide. We suggest that you continue on to the Getting started as a user guide, so you have a better understanding of how your team can useJira Core.

Getting started as a user

This tutorial shows you how to work with Jira Coreon a day to day basis. Here's what you can count on learning:

1. How to access your projects
2. How to work with issues
3. How to search for issues

By the end of this tutorial, you'll have a good understanding of how you can work with Jira Core, and be able to navigate around a project and search for issues.

Audience

General users, project administrators, administrators

Time

20 minutes

Ready to tame this beast? Get started and learn how to useJira Core.

Let's get started

Accessing a project

1. Accessing a project
2. Creating and working with issues
3. Searching for issues and filtering

Business projects can be viewed as containers that hold issues. Each project has an associated workflow, and this workflow is applied to all issues held in the project. Start working on your assigned tasks by accessing and viewing the project in which they're held.

Accessing a project

When you are given access to a project and you log in for the first time, you'll see the project listed in the **Projects** dropdown menu. Let's check it out.

1. Select **Projects** > **View All Projects**. A list of all projects you have access to will display.
2. Select **Dragon Design Tees**. You'll land on the project summary page, which displays recent activity and lets you easily track the status of issues in this project.
3. If you're participating in multiple projects, you can easily see which one you are currently working on, by selecting **Projects** and looking under Current Projects.

Viewing a project's issues

You can view and filter the issues in your project so that you see, for example, only issues that are assigned to you.

1. In your project, select**Issues**from the project sidebar. From here, you'll be able to preview issues and select preconfigured filters to change which issues are displayed.

Well done! You now know how to browse to a project, and view the project details. Next, you'll create and work with some of your own issues.

Next

Creating and working with issues

An issue is the most basic entity in Jira Core. Depending on your team and its needs, issues can represent different things. In the Dragon Design Tees project example, each issue would represent the work needed to create and complete new t-shirt designs.

Create your first issue

Let's create an issue to track the creation of a new t-shirt design.

1. Choose**Create**in the Jira Coreheader.
2. Fill out the fields using the sample data is shown below. Only the fields with*are mandatory.
   - Project: *Dragon Design Tees*
   - Issue Type: *Task*
   - Summary: *Create a contractor resources spreadsheet*
   - Description:*The spreadsheet must contain a breakdown of all contractors and specific skill sets.*
   - Leave all other fields blank or at their default values.
3. Choose**Create**to create your new issue.A confirmation message will display for a few seconds. Make note of the issue key of your new issue (e.g. DT-1).

   Create two more issues to familiarize yourself with the process.

   > **Need to create multiple issues?**
   >
   > Checking the "**Create another**" check-box on the 'Create Issue" dialog will keep the dialog open after you click Create.

Editing an issue

You can easily edit your issue to add more information, attach new files or screenshots, and more.Inline editing is the quickest way to edit an issue. To access all issue fields, including blank fields that don't appear when inline editing, you can use the Edit Issue dialog.

1. In your project, select **Issues** from the project sidebar and open the first issue you created.

2. Hover over thePriorityfield. Click **Edit** ( )to edit the field.
3. Change the Priority to "Critical" and click anywhere outside the field to save your change.

The issue will be updated immediately. Now, let's see how to start working on an issue.

Progressing the issue
Every issue has a lifecycle. In Jira Core, the lifecycle of an issue is managed by a workflow. A workflow consists of the issue statuses (e.g. To Do) and the transitions between each status (e.g. Start progress).In this step, you'll start working on your issue by transitioning it from the To Do to In Progress statuses.

1. Open the first issue you created (e.g. DT-1). The issue should be in the To Do status.
2. Select**Start Progress**. The status of your issue will be changed to**In Progress**.
3. You can then choose to stop progress, or close the issue by marking it as Done and selecting a resolution.

Notice that resolving an issue prompts you to enter more information, whereas starting progress on the issue did not. Some issue transitions have screens associated with them in the workflow associated with your project. Remember, a project can only have one workflow associated with it.

> **About workflows**
>
> JIRA Platform ships with default workflows. The workflows can be changed and customized, but this can only be done by an Administrator.

Assign the issue to another user

You can better manage your work by assigning issues to other members of your team.

1. Open one of the issues you have created (e.g. DT-2).
2. Choose **Assign** on the issue.
3. Type **emma** in the **Assignee** field and select her as the assignee from the drop-down list that appears.
4. Type (don't copy and paste) the following text in the **Comment** field, then choose **Assign**.

   ```
   Hi @emma, I've assigned this issue to you to work on this week.
   ```

   ⓘ You will notice a few things when you enter the comment:

   - When you start typing after the @ symbol, you will be prompted to choose a user: emma, in this example. Emma will be sent an email notification that links to the issue, when you save. This feature is called **mentioning a user**.
   - On choosing **Assign**:
     - The issue will be assigned to Emma with a comment added to it.
     - A link will be automatically created to the issue in the issue comment.

Awesome! You've created some issues, edited one, and assigned another to a member of your team. You're really getting the hang of working with issues. Now, you'll learn how to best search for the issues you need to work on.

Next

Searching for issues and filtering

1. Accessing a project
2. Creating and working with issues
3. Searching for issues and filtering

Creating issues is important, and its equally as important to be able to search and manage multiple issues to ensure you and your team work together well.In this step, we will show you how to work with multiple issues. You will learn how to use different search techniques to find issues. We will also show you how to share search results with your team and report on issues.

Create some more issues

Before you start, you are going to need a few more issues. Create a few more in yourJira Coreproject, using the sample data below.

- **Issue Type**=Taskand**Summary**=Brainstorm new designs
- **Issue Type**=Taskand**Summary**=Approve new t-shirt design
- **Issue Type**=Task,**Summary**=Send mockup to printersand**Assignee**=kate

> **Want to create issues in bulk?**
>
> If you're familiar with the CSV format, you can create a CSV file to import issues in bulk. This can be handy if you're handling a lot of data.

Search for issues

In this example, we are going tackle a common scenario: searching for all unresolved issues assigned to you. You might regularly run a search like this to check your backlog of work.

1. From your Jira CoreCloud site header, select**Issues**>**Search for Issues**. You should see issues from your demo project and any other projects you have access to.
2. Set**Assignee**=**Current User**in the search criteria.

   ⓘNotice that the search results refresh when you select new criteria.
3. Choose**More**> type**Resolution**then select it.
4. Set**Resolution**=**Unresolved**. The search results will show the issues that are unresolved and assigned to you.

If you are thinking that it would be handy to be able to rerun this search, we have got you covered! Hover over the expand icon (»)icon in the top left and choose**My Open Issues**. Keep this screen open for the next step.

Save your search

If you run a search with the same criteria frequently, you may want to save it as a**filter**. This lets you run the search again with a single click, rather than selecting the same criteria every time. For example, you may use a filter to review your open tasks for the day.

In this step, you will search for all tasks assigned to Kate in the Dragon Design Tees project, and then save this search as a filter.

1. Select**Issues**>**Search for Issues**to start a new search.
2. Set**Project = Dragon Design Tees**and**Assignee = kate**as the criteria. You should see at least one issue.
3. Select**Save As**(above the search criteria), enter**Kate Dragon Design issues**as the**Filter Name**and save it.

That's it! Hover over the expand icon (»)in the top left. You can see your new filter under the**Favorite Filters**section. Just click it to run it.Let's now look at some of the ways that you can use your new issue filter.

Share your search results

Getting your team on the same page is easy with shared filters. You could share a filter with your team that shows the unresolved stories for a development iteration, or the critical issues in a support backlog.

Here are two ways that you can share search results:

Email the search results

Run the desired filter, then choose**Share**. Enter the users that you want to share the filter with and they will be emailed a link to your filter (if you have email notifications set up).

Share the search results via a dashboard

The dashboard is the screen that all Jira Coreusers see when they first log in. You can show a filter's results on a dashboard and share it with other users.

1. Choose**Dashboards**>**Manage Dashboards**, then choose**Create new dashboard**.
2. Name your dashboard**Dragon Design Tees**and choose the**+Add**button next to**Add Shares**to share it with everyone.
3. Leave the other fields and choose**Add**.
4. Choose**Dragon Design Tees**in the**Favorite Dashboards**section to configure it.
5. Choose**add a new gadget**to open the 'Gadget Directory'.
6. Enter filter results in the search box and choose**Add It Now**.
7. Enter**Kate Dragon Design issues**in the**Saved filter**field and choose**Save**.

Other users can now add this dashboard by choosing it as a favorite.

That's it! You've now performed some searches and filters, and learned to to save and share them with your team. You're ready to jump right in and experience the full power of Jira Core!

You've now completed the Getting started as a user tutorial. For more information on usingJira Core, continue to theJira Core[documentation home](#).

Administering a project

Working in Jira Coreis all about working with issues in projects.AJira Coreproject is a collection of issues, and issues are the basic packets of work that need to be done.A project allows you to apply a process to each issue (via an associated workflow) within your project. You could choose to create a project to track leave requests, or a project to create and monitor a marketing campaign, Jira Coreallows you to modify your project to suit your needs.

Project details	A project must have a name, a project key and a project lead. These details help identify the project and the issues within it. *Learn more: Editing a project's details*
Project access	Access to your project is controlled through project role membership. You can assign individual users or groups to the project roles. *Learn more:Managing project role memberships*
Versions	A version is a way to group issues within a project. Versions have a start and end date, and can be used to effectively group issues (and therefore your work) into deliverables by date *Learn more:Organizing work with versions*
Components	A component is another way to group issues within a project. Components don't have dates assigned to them, but they can have a component lead who could be a subject matter expert, or a team lead for that area of the project. *Learn more:Organizing work with components*
Workflows	Each project has at least one workflow that can be applied to it's issues. A workflow controls how an issue progresses, from creating the issue to closing the issue. *Learn more: Workflows*
Issues	Issuesare the packets of work that need to be done, and each issue contains information held in fields. You can customize your issues by changing the fields, to make sure you always have the information required to complete your work. *Learn more: Customizing the issues in a project*

Editing a project's details

You are able to edit the project name, description, project category, avatar and URL of a project that you have the Project Administrator permission for.

Note that when you change the name of the project, this change *will not be reflected in any saved filters* that contained the project's name and they *will be affected* by the change unless your filters are using the project ID or the project key (this too if it is unchanged).

Editing a project

1. Navigate to the administration page for the project:
   - Choose **Administration** (⚙) **> Projects**, or
   - Navigate to the desired project's summary and click the **Project settings** button at the bottom of the project navigation sidebar.
2. Select **Details**.
3. Make your edits.
4. Select **Update**.

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, seeManaging project roles.

Viewing and editing project role members

1. Log inas aproject 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.

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.

You need to have the project-specific**Administer Projects**project permissionor the**Jira Administrator**global permissionto 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** (ϕ)>**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

The**Versions**page is a visual representation of how your versions are progressing. The data represented is taken from the Fix Version/s field,and shows the status of the issues assigned to that version.

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:
   - simplenumeric, e.g. "2.1", or
   - complicatednumeric, e.g. "2.1.3", or
   - aword, 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.thepla nned release date for a version) can be 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**fro m 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.

> (i) To revert the release of a version, simply select**Unrelease**from the drop-down menu.

Archivea version

1. On the 'Versions' page, hover over the relevant version to display the cog icon, then select**Archive**fro m 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.

> (i) 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.Versi onsselectedonthis 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.

Organizing work with components

Components are sub-sections of a project. They are used to group issues within a project into smaller, more manageable groups. You can set a default assignee for a component, and this will override the project's default assignee for issues in that component.

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".

Componen t 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.

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

- Edit existing workflows - Create new workflows - Configure existing workflows	Working with workflows
- Add a workflow scheme - Configure a workflow scheme - Manage workflow schemes	Configuring workflow schemes
- Import and export workflows - Activate and deactivate workflows	Managing your workflows
- 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

Working in a project

A project inJira Coreis a collection of issues.Your team could use a project to coordinate the development of a product, track a project, manage a help desk, and more, depending on your requirements. A project can also be configured and customized to suit the needs of you and your team.

This section of the documentation covers working in a project that's already been set up for you.

User profile	Working in a project, andJira Corein general means you can log in. And if you can log in, you have a profile that you can edit to make sure it's configured how you want it to be configured. *Learn more*about how to manage your profile
Viewing a project	Getting to grips with how your project looks and is set up will help you work more effectively and efficiently. *Learn more*about how to view a project and its contents
Working with issues	Issues are the packets of work that need done in a project. Knowing what you can do with them, and how to do it, is the basis of all work inJira Core. *Learn more*about working with issues
Searching for issues	You can't work on issues if you can't find them. Working out what the most effective way to search for issues is, and how to use the powerful advanced search function, will help you always be able to locate your issues when you need them. *Learn more*about searching
Reporting	Reporting helps you manage and track your work. *Learn more*about reporting inJira Core
Gadgets	Gadgets are nifty ways of displaying data about your project and your issues. Setting them up and using the right ones makes sure you always have your stats at hand. *Learn more*about gadgets
Dashboards	You can set up dashboards, and share them with team members, so that everyone is viewing the same information on your projects and issues. You can add gadgets to dashboards, and create wallboards which allow you to display your stats anywhere you can connect to. *Learn more*about dashboards and wallboards

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**.

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 edit icon ✎ 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**MyJiraHome**section:
    - Dashboard
    - Issue Navigator
    - Boards (available if you're using Jira Software)

    ℹ️Your page will be reloaded theJirahome page you selected.
3. *(Optional)*To verify that yourJirahome page has been reset, log out and log back in toJiraagain. You should be taken directly to theJirahome 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, seeTranslating Jirafor more information.
- Your **time zone** specified in your profiledoesn'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'skeyboard 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 givegadgets access torestricted dataon an external, OAuth-compliant web application or website (also known as a "consumer").Check outAllowing 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 you 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

YourJiraadministrator must link your Jira instance and the consumer using an application link and OAuth. For example, if you want to add aJiragadget to your Bamboo homepage, then yourJiraadministrator must first approve Bamboo as an OAuth consumer.

> (i) 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 Jiragadget 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 Jiragadget access to your Jiradata. Whenyou revoke access, the gadget can only access public data on your Jirainstance.

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 yourinstance, you can submit requests for Marketplace apps directly to youradministrator.

The 'Atlassian Marketplace for Jira' page presents an integrated view of the Marketplace website from within the Jirauser 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.

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 additionalkeyboard 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 theKeyboard Shortcuts dialog box.

1. Ensure you are logged inand open the Keyboard Shortcuts dialog box (seeabove).
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 thePreferencessection of your user profile. See Managing your user profilefor 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 readMozilla's documentationfor 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	

77

Viewing a project

When you select a project to view for the first time, you're taken to the project Activity page. If you've viewed the project before, you'll be taken to the last screen you visited on it. The Activity page provides a summary of your project activity, and further navigation which allows you to 'dig down' into further detail. The sidebar provides navigation to all the issues in the project, and the reports available. If you're a project administrator, you can also add unique shortcuts to information and other webpages to the sidebar, which are accessible to all users with access to the project.

You can access the project Activity page by selecting the **Project** drop-down, and selecting your project from the list. If your project is not listed, select **View all projects** to search for your project. Once you're viewing the project, click the **Summary** link to view the Activity page.

From the project summary screen, you can view the following by selecting the link in the project navigation sidebar:

- **Project:**
  - Summary (as shown above) Shows recent activity in your project, the project lead, the most active users within your project, and the project key.
  - Issues Takes you to the issue navigator, which shows a list or detailed view of issues in your project.
  - Reports Shows reports on statistics for particular people, versions, issues or other fields within issues.
  - Components* Shows a summary of all components for a given project.

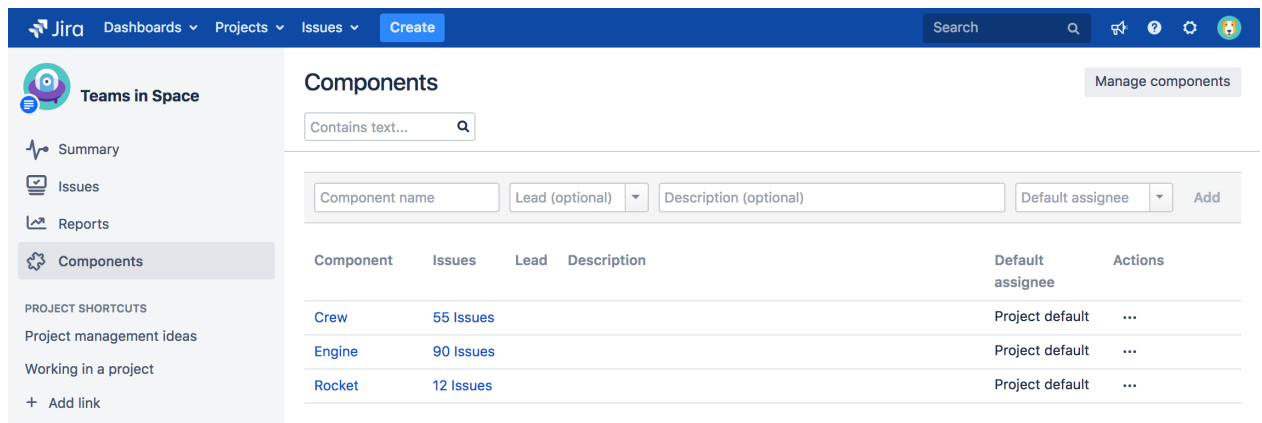*Components are only available if your project administrator has created versions or components within the project.

- **Project Shortcuts:**
  - 2023-07-11_15-32-59_Project shortcuts can be added to your project navigation page to any online resources your team may want to access. These shortcut links are available to everyone who has access to the project.

Viewing a project's components

Jira'sComponents page shows a summary of all components (if any have been created) in a project. You can search for components by text contained within the component's name or description.

To browse a project's components,

1. On the top navigation bar, click the white triangle next to **Projects**. The projects drop-down will display.
   ✅ **Tip:** You can access your current project directly by simply clicking the **Projects** link instead of the triangle.
2. Click the project you wish to browse. If the project is not displayed in the drop-down, click **View All Projects**, which allows you to view a list of all accessible projects on your JIRA instance, and select your project from there.
3. Click **Components** on the left of the page. A list of components for your project will display (see screenshot below).
   - Click the name of a component to view all the issues related to the component.

Viewing a project's issues

The Activity page allows you to access the issues contained in that project via the project navigation sidebar. The issues are displayed as a list, and have several filters you can select to view the issues relevant to you.

To view a project's Issues,

1. On the top navigation bar, click the white triangle next to **Projects**. The projects drop-down will display.
   ✅ **Tip:** You can access your current project directly by simply clicking the **Projects** link instead of the triangle.
2. Click the project you wish to browse. If the project is not displayed in the drop-down, click **View All Projects**, which allows you to view a list of all accessible projects on your JIRA instance, and select your project from there.
3. Click the **Issues** tab in the project navigation sidebar. An embedded issue navigator is shown for all the issues in your project.

   Tip: To view an expanded view of an issue, click the expand (↗) icon on the issue or any issue link in the issue (description or comment).

Working with issues

Need help working with issues?Jira Coreallows you to create issues quickly, assign them to the right person, and get working on them in seconds! On this page, you'll find a quick overview for everything that you can do with an issue, as well as links to pages with more detail.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.

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 Core, an issue is essentially a packet of work. It could be a small task like "Remember to order pizza for charity night", or a large chunk of hard work like "Build bridging wall between house and garage"! It all depends on your project, and how you and your team decide to break down your work into issues.

An issue is broken down into several key areas. Here's an example of an issue send out an invoice, you can see all the critical information, such as the assignee, due date and description, all in one place.

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 team, you can attach documents, images, and screenshots to your Jira application issues.

Multiple files can be attached to an issue. Click an image thumbnail to open a preview, and if there is more than one image in the gallery, navigate to the next image preview by clicking the right arrow in the preview.

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 outConfiguring 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.

Add attachments

You can add files and images to any issue in your service desk project. When working on an issue, simply drag and drop a file onto the issue, or select**More**>**Attach files**. You will then have the option to add a comment with more information about the attachment, and then 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.

You can add file and image attachments to any issue.To add an attachment when you first create an issue, copy a file from your computer and paste it directly in the**Create Issue**dialog.

> ⓘ Make sure that you paste the image outside the **Description** box.

To add attachments to an existing issue, open the issue and follow these steps:

1. Click**More >Attach files**.
2. Add a file(s).
3. Click**Attach**or**Open**.

You can also drag and drop files onto an issue to attach them.

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.
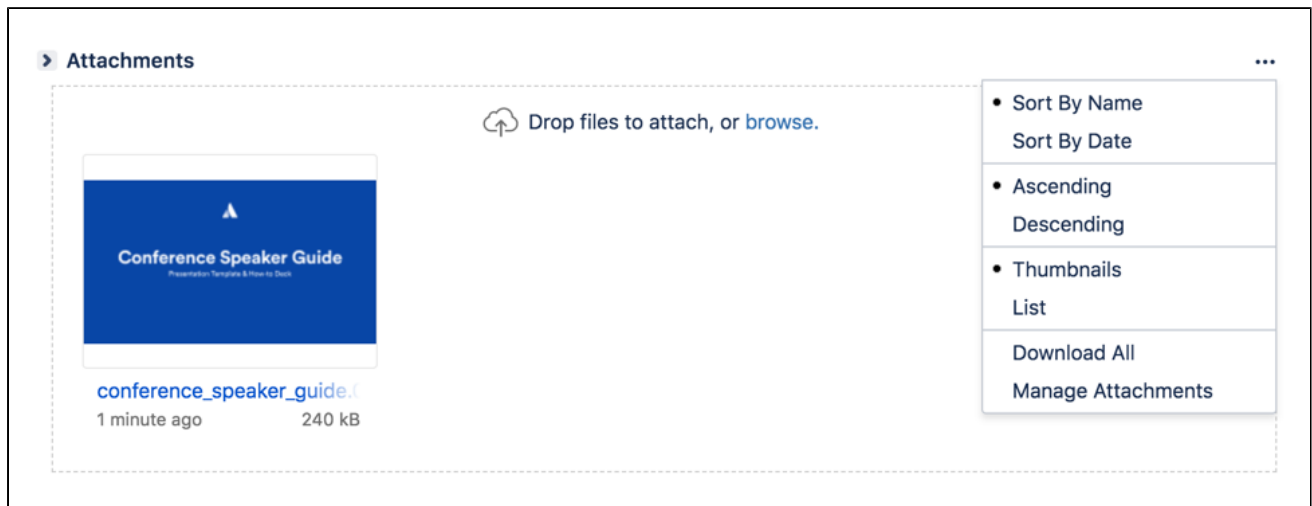
Access ZIP file contents

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.

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.

Creating a linked issue

From any existing issue you're working with, you can create a linked issue between Jira Software projects or Jira Software and Jira Service Management projects.

The linked issue will relate to the existing issue in a way you select. For example, the linked issue might block the existing one or be resolved by it. You can configure fields in the linked issue directly on the issue screen.

To create a linked issue:

1. Open an issue to which you want to link a new issue.
2. Select **More** > **Create linked issue**.
3. In the dropdown list, select **Create linked issue**. The **Create linked issue** screen will appear.
4. On the screen, the default and custom fields reflect the fields of a destination project. The fields on the linked issue screen are prepopulated with the data from the original issue youre viewing. You can edit themas needed.
5. After you finish working with the fields, select **Create**.

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.

> (i) You can only create sub-tasks if your administrator hasenabled sub-tasks, and has added the sub-task issue type to the project'sissue 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 forJira Softwareissues)

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 toIssue**.
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 levelfor an issue, your administrator must add you to the appropriateissue security level, and also grant you the'Set Issue Security' permissionfor 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 levelsyou 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 levelwill 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.

> ⓘ 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

> ⓘ 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 shouldnt contain any punctuation, except for commas separating columns or fields. Otherwise, the import may work incorrectly.

- Commas as column or field separators cant 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, youll 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 theprojects to which they will be imported.

> (i) If you arent a Jira administrator,you wont be able to map the fields **Project Name** and **Project Key** to Jiras fields. Youll need to select different fields for mapping. Learn more about this in Ti ps 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 arent a Jira administrator, you wont be able to map the **Worklog** field to Jiras field. Youll 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. Heres 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 Jiradoes 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

> ⓘ 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** toopen 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.
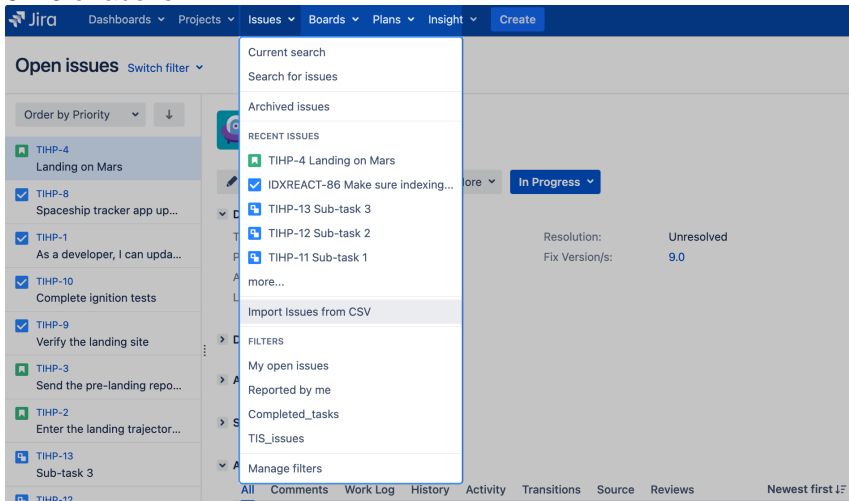
2. On the **Bulk Create Setup**page, select your**CSV Source File**.
   Leave the**Use an existing configuration file** checkbox cleared if you do not have a configuration file or if you want to create a new configuration file. Configuration files specify a mapping between column names in your CSV file's header row and fields in Jira.
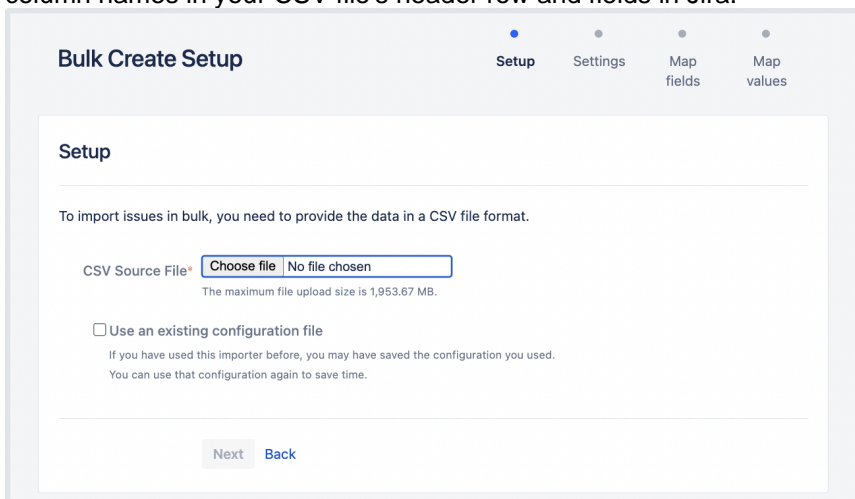
> ⓘ If you select **Use an existing configuration file** checkbox, you will be asked to upload an **Existing Configuration File**. If you don't select this option, Jira will automatically create a configuration file that you can use for further imports.

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.

5. 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**.

> ⓘ 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.

6. 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**.

> ⓘ - 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 **Assig nee**, usernames from the file will be automatically mapped to existing usernames (lowe rcase) 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.

> (i) 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 givem 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, deliminated 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	Choose to automatically create Jira users for any values of the Assignee or Reporter field. • 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 wont 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. Youll see the list of users that can't be created. • If Assignee and Reporter aren't mapped, no usernames will be created.
Other fields	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. If your custom field is a date field, please use the date format specified n the second step of the CSV import wizard: `dd/MMM/yy h:mm a`.

> ⓘ If you arent a Jira administrator, you wont 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

Complete your work more efficiently with these tips and tricks for editing and collaborating on 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 mentioning other people
- Keeping track of issues with labels and issue watchers

Attaching files and screenshots

If your administrator has enabledfile attachments, you and your customers can attach files and screenshots to issues you're working on. SeeAttaching files and screenshots to issues for more information.

Collaborating on issues

Sharing issues with other users

You can easily keep your team informed by sharing issues with them via email.

To share the issue:

1. In the upper-right corner of the screen, select the **Share** button.
2. Enter usernames or emails of users that you want to share the issue with.
3. Select **Share**.

If your administrator has enabled anonymous access to your project, you can also share issues by entering the email address of a non-Jira user.

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.

Mentioning users on issues

If you want to invite members of your team to help you work on an issue, you can mention them by typing @ and their usernames 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.

Editing issue details

> ⓘ **What permissions do you need?**
>
> To edit an issue, you need the**Edit Issue**project permissionfor 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 theEdit Issuedialog 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.
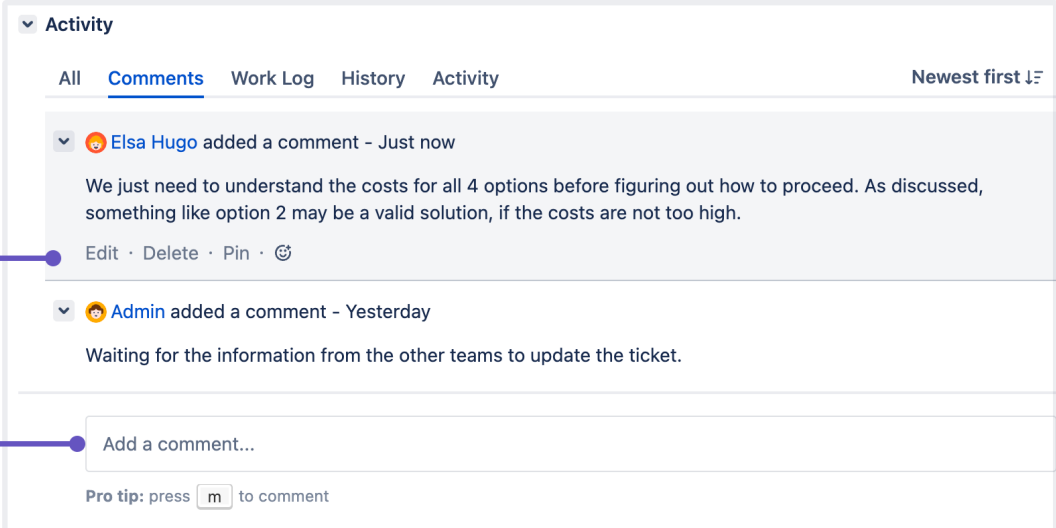
Commenting on issues

> ⓘ **What permissions do you need?**
>
> To add comments to an issue, you must have the following permissions for the relevant project:
>
> - **Browse project**permission to view the issue you want to comment on.
> - **Add comments** permission to add a comment to the issue. Without this permission, you wont see the **Comment**button.

You can add and manage comments in the issue view.

1. Edit, delete, or pin comments.
2. **Sticky comment footer**: Add a new comment to an issue.
3. **Sort comments**: display comments from newest or oldest first, or vice versa.

Add a comment

Open the issue you want to comment on and add a comment in the way you prefer:

- Start typing your message in the sticky comment footer.

- Press **m** to jump straight to the comment footer and start typing your update.

In Jira, you automatically become a watcher of the issues that youve commented on. To disable this:

1. Select **your user name** in the upper right corner of the screen, then select **Profile**.
2. In the **Preferences section**, set **Autowatch** to **Disabled**.

Delete a comment

Open the issue youve commented on, find the comment you want to remove, and then select **Delete** at the bottom of the comment. Confirm that you want to remove this comment from the issue by selecting **Delete** when prompted.

> ⓘ You cant delete comments added by other users.

Edit a comment

Open the issue, select **Edit** at the bottom of your comment, and edit the text or restrictions (**Viewable by...**) as needed. When you save your revised comment, youll see the text edited next to the comment date, indicating that the commment 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 y our 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

Apply viewing restrictions to a comment by selecting the open padlock icon 🔓▾ (or🔒▾ Restricted to Users if restrictions already apply).

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:               ☑ 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 theLabels Gadgetto your dashboard to quickly find issues with labels relevant to you and your team.

Watching and voting for issues

> ⓘ **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 thevoting 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.

Issue links within an issue look like this:
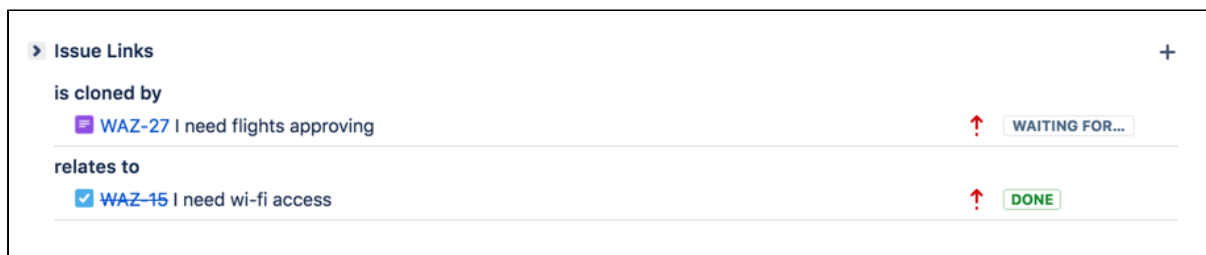
Note: Resolved issues (i.e. issues with a Resolution set) are displayed in strike-through font, e.g. ~~DEMO-1~~.

> ⓘ 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 selectedat the left of the dialog box and then choose the type of link to be created from the**This issue**drop-down list.
   ℹ️If your Jira system administrator has configured*fully reciprocal application links*between your Jira site and another one, a**Server**drop-downlist 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.
     ℹ️If 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 selectedat the left of the dialog box.
   ℹ️**Note:**
   - This option will not be available if your Jira system administrator has not configured anapplication linkbetween your Jira site and the remote Jira site.
   - If, after selecting this option, you are prompted forauthorization, you may be required to log in to the remote Jira site, which will allowyourJira site to access the remote Jira site*on behalf of your account on the remote Jira site*.
     ℹ️This behavior means the application links configured between your Jira site and the remote Jira site use OAuth authentication.
4. If your Jira siteis connectedto multiple remote Jira sites, choose the relevant Jira site from the**Server**drop-downlist.

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 accesson 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 typeto a remote issue, the reciprocal link generated on the remote issue will be a**is blocked by**link type backto 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 remoteissueback to yourissue 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 reciprocalapplication 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 createdasthe 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),

thena 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 theLink 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 theauthorizationstep (at step 3),authenticateon the remote site with a user account which has this permission.
  If you are not prompted for authentication duringauthorization, 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

To create a linked issue, you need to have Create issue and Linked Issuespermissionsin the destination project(s).

To create a linked issue:

1. Open the issue from which you wish to create the linked Jira issue.

2. Inthe 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.

3. Select the destination **Project** in which the new linked issue is to be created.
4. Select the correct Issue Type for the new linked issue.
5. In the**Linkedissues** field, specify issue(s) to be linked to your new linked issue.
6. Edit the linked issue**Summary.**
7. Edit the **Description** anddescribe why you are linking these issues.
8. Select the**Copy attachments**checkbox to include any attachments from the original issue.
9. Select the**Copy links**checkbox to include any URLs from the original issue.
10. 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*betwe
en your Jira site and the Confluence site containing the pages 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. Click the**Confluence Page**option at the left of the dialog box.
   ℹThis option is not available if your Jira system administrator has not configured anapplication
   linkbetween your Jira site and Confluence site.
4. 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-downlist.
5. 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 forauthorization, you may be required to log in to the Confluence site, which will allowyourJira site to access the Confluence site*on behalf of your accountonthe 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:**IfConfluence pagelinks you create show**Failed to load**on theissue 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.**'

thenJira was unable to communicate with the Confluence server to either:

- retrieveinformation about the link or
- conducta 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 behype rlinkedto 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 forlinkedissues

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 appropriateproject-specific permissionand the globalBulk Changepermission. 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.

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 orproject administratorof all the projects associated with your selected issues.

> ⓘ **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 thenn 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, youll only be asked to move sub-tasks if youre moving a parent issue to another project. This is to maintainthe parent/sub-task relationship (i. e.thesub-task is always located in the same project as the parent issue).For example, you haveissueB 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 youdon'tchange 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**.

My open issues	Save as			Share	Export ▼	Tools ▼

✓ assignee = currentUser() AND resolution = Unresolved order by updated DESC   ? **Search**   Basic

Bulk Change:
all 16 issue(s)

Clear Sorts
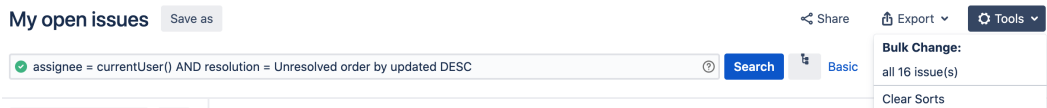
3. Select the issues you'd like to perform the bulk operation on, and select**Next**.The bulk move operation may require additional information depending on which issues you have selected to move (expand the following dropdown for details).

Select Projects and Issue Types	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. 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.
Select Projects and Issue Types for Sub-Tasks	If you are moving issues with sub-tasks to another project, you will also need to move the sub-taskstothe new project. You can also elect to change the issue types of the sub-tasks being moved if you need to.
Selectstatus migration mappings for invalid statuses	As multipleworkflowscan 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.
Select values for required fields and fields with invalid values	To adhere to thefield configuration schemeassociated with the target project and issue type, it may be necessary to update/populate required fields (e.g.fieldsthat 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 movedtogether. For the following fields, you can select from a list of possible values provided for you: • Component • Affects Version • Fix Version • Custom fields of type "Version-Picker" ⓘ Versionswhichhave been archivedin 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 firstunarchivethe version in the target project. 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. • **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:

- 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 migrationmappings 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.targetproject, 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 editoperations 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**.
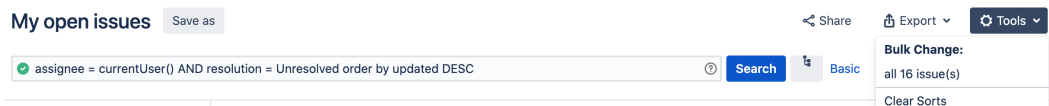
3. Select the issues you'd like to perform the bulk operation on and then select**Next.**
4. Select **Edit Issues**and then select **Next.**
5. Select the bulk edit operation from the list of available operations (expand more information for a full list of available and unavailable operations, and their conditions).

Available Operations	Conditions
Change Affects Version/s	• 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	• 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	• 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	• 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	• 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 "schedule issue" permission for all the selected issues
Change Fix For Version/s	• 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	• Current user has "edit issue" permission for all the selected issues
Change Priority	• 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	• 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	• 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	The "Change Custom Fields" operation is available only if: • 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	• Your workflow must allow editing of closed issues
Change Sprint	You need to specify the sprint ID. • 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 Trackingfields: Original Estimate, Remaining Estimate, Time Spent

The following custom field types are unavailable for bulk editing:

- Import Id
- Read Only Text
- a. Select a value for any required fields for this operation, and if available, decide whether you'd like to send email notifications. Select **Next**.
- b. 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 of1000 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 of1000 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 Issuespermissionfor the project that has the issue that you want to move.
- You must have theCreate Issuespermission for the project that you wish to move your issue to.

If you do not have either of this 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

TheMove issuewizard 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, theMove issuewizard allows you to change certain attributes of the issue. These include:

- **Issue type** If your issue is acustom issue typethat 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 upcustom issue statusesas 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 fieldsfor 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 theMove issuewizard 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.

Visual	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 lik • color • underline • emoticons 🙂 and I even get **Tables** that look like tables!
Text	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|| |that look like tables!|

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 it's 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 it's content will be rendered as text (wiki markup).

Scheduling an issue

You can schedule issue due dates in Jira Core to track and review, and inform teams about issues dates. 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.

> (i) 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

TheDue 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 seeAdvanced Searching, and particularly the documentation on theDuefield.

Logging work on issues

Jira Core provides the flexibility to set your estimation and tracking statistics differently depending on your team's needs. Time Tracking features projects schedule planning and time expectations management thanks to its reports, which show original and current time estimates for all the issues, and whether they are ahead or behind the original schedule. Teams often need to be able to estimate how long a product will take to deliver in order to have conviction that the work will be completed. You will be able to easily track the tracking time info by just watching the issue time tracking color bars.

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
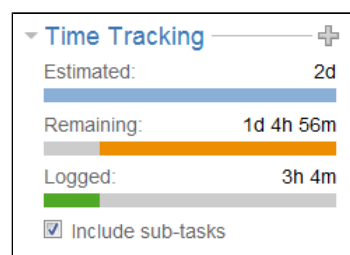
When you log time for the first time, thetime spentis subtracted from theoriginal estimate, and the resulting value is automatically presented in theremaining estimate. When subsequent work is logged, anytime spentis subtracted from theremaining estimate.

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 Logsproject permissions.

  > (i) Note that anyone with theBrowse Projectpermission 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 administratorto disable legacy modeon 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'), thedefault time unitthat 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 estimated	Amount of time anticipated to resolve the issue after completing this unit of work. You can adjust this value using the following options: • **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	Type a description related to the achieved work.
	Commentsare copied to the**Workflow Description**by default, but yourJira administrator can change this option in the'Copy Comment to Workflow Descriptions' settings. If this setting is disabled: • 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. Selectthe 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 canedit your own work log entries if you have been granted the Edit Own Work Logspermission.You can also editother 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 issueremaining estimatevalue.
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 andspecify time estimates on the same Jira screen when performing any Jira operation, such as editing, creating an issue, ortransitioning 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**fiel ds 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.

117

- If your Jira administrator hasconfigured 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.

Customizing the issues in a project

Issues are the packets of work that need to be completed in a project. They are made up of issue fields, and these fields contain important data about the issue such as a summary, description, due dates, and when and where the work is required. This information is presented on a screen.

A screen groups together all available fields (or a subset of fields) and organizes them for presentation to a user. Learn more about Jira screens

Options for customizing issues

Jira Coreallows you to customize issues by changing the configuration and behavior of these fields and screens to suit your team's needs.

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 custom fields. Learn how to add new fields to Jira
- Configure different renderers for some fields. Learn more about field renderers
- Change the position fields on a screen
- Choose which screen should be displayed for each issue operation (e.g. create or edit issue) or workflow transition (e.g. resolve or close an issue)

A simple example of how customizing an issue could benefit your team could be marking fields as "Required" when an issue is created. Learn how to configure fields behavior

By using required fields, you always capture the necessary information you need to get the work done and resolve the issue. If you couple this functionality 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 that your users can see and complete the required fields as quickly as possible.

You can turn this...

into this...

> ⓘ Jira administrators and project administrators have different permissions when it comes to customizing issues. Learn more about permissions in Jira

How project administrators can configure issues

As a project administrator, you can 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 that you modify arent used by other projects or 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 with another project, you'll see this information when you view the screen.
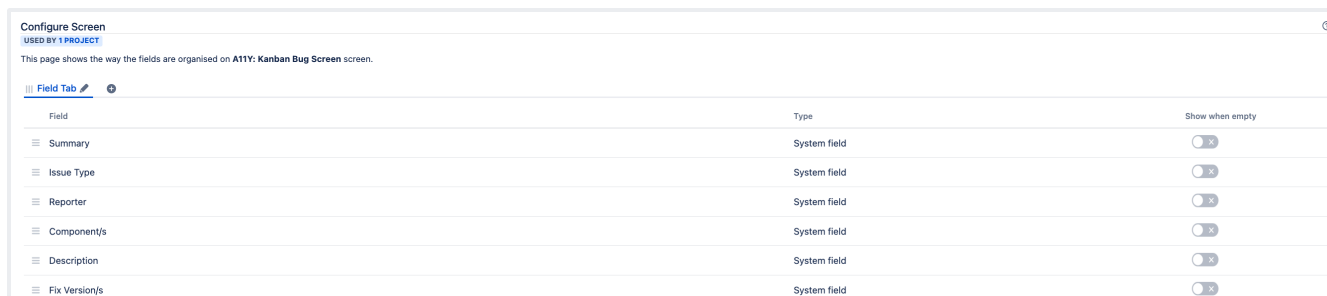
If your screen suits the preceding criteria, you can:

- add and remove tabs that will appear on a screen, as well as edit the name of the tab
- add, remove, and rearrange system fields
- add, remove, and rearrange existing custom fields, but not create custom fields

Editing screens for issue operations

1. Open the **Projects** dropdown and choose the project whose screens you want to manage
2. In the project sidebar, select **Project settings**
3. In the **Project settings** sidebar, select **Screens**
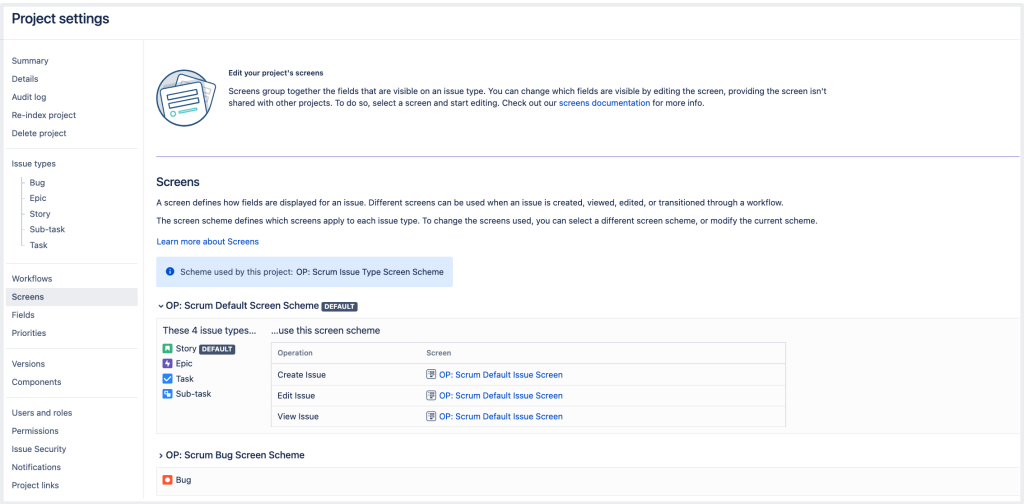
You'll see the screen schemes used by your project, and the issue types that use that scheme.

To view the screens associated with that scheme and issue types, expand the screen scheme section. Select the screen name to open the screen configuration.
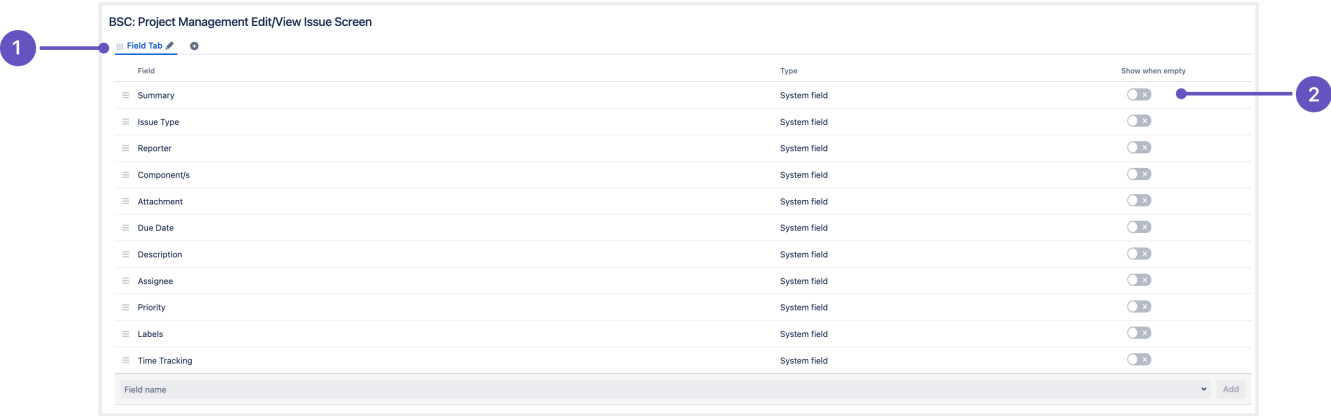
1. **Field tab** this is how you can group related fields together.
2. **Show when empty** toggle when enabled, empty custom fields will be visible in the issue view.

> (i) Alternatively, you can configure project screens from the **Issue types** tab in the project settings. After you select the issue type you want to configure screens for, youll see the same screen configuration page.

You can perform the following operations on the screen configuration page:

Action	Instructions
Add a tab	Select**Add tab** ⊕ . Enter a new tabs name in the dialog that appears and then select **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**.

121

Add a field	1. Select 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. Select **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.
Display an empty field in the issue view	To make a custom field with no data visible in the issue view, turn on the **Show when empty** toggle. To hide an empty field from the issue view, turn the toggle off. Similarly to system fields, Jira will automatically set the None value for emty fields.

How Jira administrators can configure issues

As a Jira administrator, you can view more conceptual informationoncustomizing issuesin the Jira administrator's documentation:

- Configuring issues
- Defining a screen

Searching for issues

Can't find the issue that you are looking for? This page will show you how to search for issues in Jira. Anyuser can search for issues, although they will only see issue results from projects where they can view issues (i.e. 'Browse Project' permission).You'll find a step-by-step guide below that will show youhow 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.

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 thequick search, using thebasic search, or using theadvanced search.

Quick search	The quick search is the fastest way to define search criteria. However, it is less precise than oth complex queries (e.g. `project = Jira AND status = Open AND priority = High`). your search criteria is not complex, for example, you know the project key and some key words **To use the quick search:** Enter your search criteria in the search box in the header bar of Jira *Tip: If you know the issue key or project key, enter it before other search terms, e.g. "JRA help*
Basic search	The basic search is more precise than the quick search, but easier to use than the advanced se user-friendly interface that lets you define complex queries, without needing to know how to use searching). **To use the basic search:** Navigate to **Issues** (in header) > **Search for issues**, then enter you *Tip: If the advanced search is shown instead of the basic search, click **Basic** next to the 🔍 ico* Search Save as ⏴ Share Teams in Space ⌄ Epic, Story ⌄ To Do, in prog... ⌄ Assignee: All ⌄ Contains text More ⌄ **Sear**
Advanced search	The advanced search is the most powerful of the three search methods. You can specify criteri defined in the other searches (e.g. `ORDER BY` clause). However, you need to know how to cons queries using the Jira Query Language (JQL) to use this feature. **To use the advanced search:** Navigate to **Issues** (in header) > **Search for issues**, then enter *Tip: If the basic search is shown instead of the advanced search, click **Advanced** next to the 🔍* Search Save as ⏴ Share ✅ project = TIS AND issuetype in (Epic, Story) AND status in ("To Do", "in progress") ⑦

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.
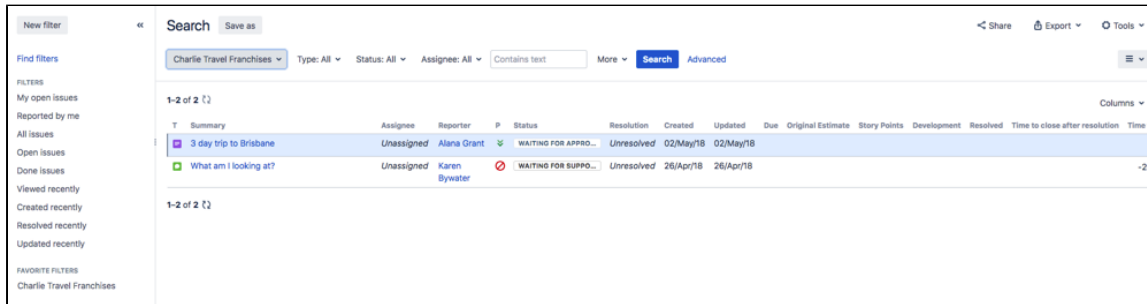
3. Working with the search results

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 searchinstead.
- If you are comfortable with the Jira Query Language (JQL), you may want to useadvanced searchinstead. This search is more powerful than than the basic search.

Screenshot: Basic search
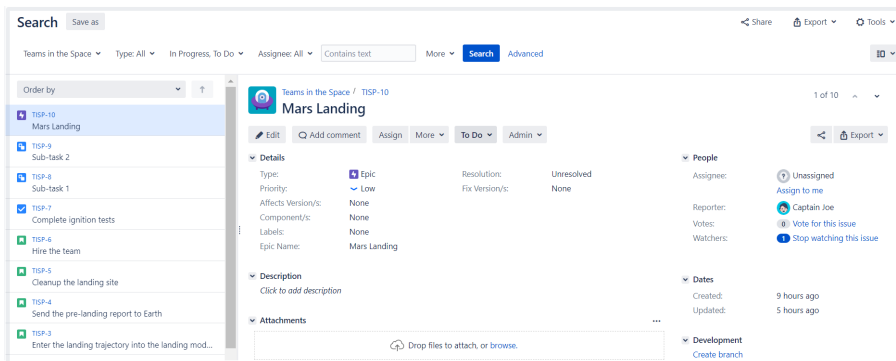
Basic searching

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 theTroubleshooting sectionbelow.
   - If you are searching for text, you can use special characters and modifiers in your search text, such as wildcards and logical operators. SeeSearch 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.

New filter	«
Find filters	
FILTERS	
My open issues	
Reported by me	
All issues	
Open issues	
Done issues	
Viewed recently	
Created recently	
Resolved recently	
Updated recently	
FAVORITE FILTERS	
Current sprint	...
High-priority issues	...
Teams in Space	...
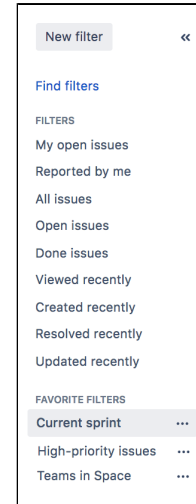
Troubleshooting

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.
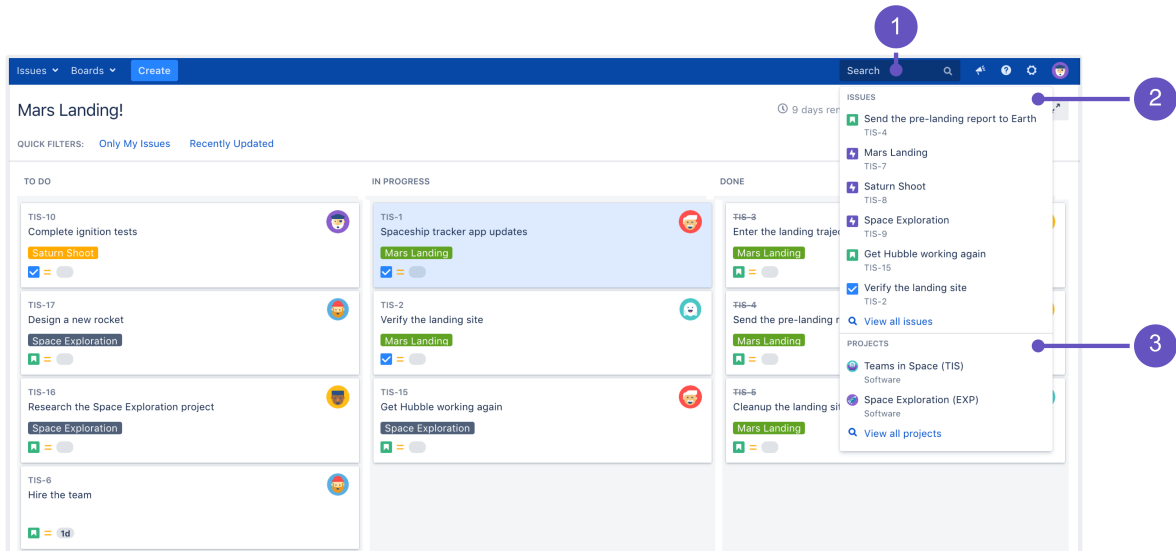
1. **Search:** Click anywhere in the box to display your recent work, or start typing to search through all your issues and projects.
2. **Issues:** Recent issues (before searching), or issues that match your search.
3. **Projects:** Recent projects (before searching), or projects that match your search.

> ✅ 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

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 youre 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 you wildcards, e.g. "`win*8`".

For more information on free-text searching, see Search syntax for text fields.

Using 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 theIssue 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 *r:* followed by a specific reporter term, such as *me*, a username or *none*. *Note that there can be no spaces between "r:" and the specific reporter term.*	`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 *created:*, *updated:*, or *due:*, respectively. For the date range, you can use *today*, *tomorrow*, *yesterday*, 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`

`<resol ution>`	Find issues with a particular Resolution.	`fixed`
		`duplicate`
		`cannot reproduce`

`c:`	Find issues with a particular Component(s). You can search across multiple components. *Note that there can be no spaces between "c:" and the component name.*	`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 `'*'`. *Note that there can be no spaces between "v:" and the version name.*	`v:3.0` finds issues that match the following versions (for example): • 3.0 • 3.0 eap • 3.0 beta ...but will not match against the following versions (for example): • 3.0.1 • 3.0.0.4 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.
`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): • 3.2 • 3.2-beta • 3.2.1 • 3.2.x

> ⓘ In Mozilla-based browsers, try creating a bookmark with URL `http://<your-Jira-site> /secure/QuickSearch.jspa?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...".

Disabling smart querying

If you dont want to use smart query as a default search behavior, you can disable it in your User profile, in the **Preferences** section.

In the **quick searching** setting, select **Text** as a quick searching mode. Jira will no longer update your search strings and will use the exact search strings to find what youre looking for.

1. **Quick searching** preference where you can select a default mode for quick search.

Searchingissues from your browser's search box

If you are using Firefox or Internet Explorer 8 (or later), you can add your Jira instanceas a search engine /provider via the drop-down menu next to the browser's search box. Once you add your Jira instanceas 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.

> ✅ Learn more about searching in Jira from JQL: The most flexible way to search Jira (on the Atlassian blog)

On this page:

- Using advanced search
- Reference
- Running a saved search
- Notes

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.
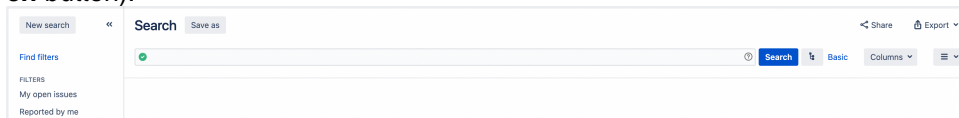2. A list of Jira **issues** that match the search criteria.

Using advanced search

1. Go to **Issues** (in the header) > **Search for issues**.
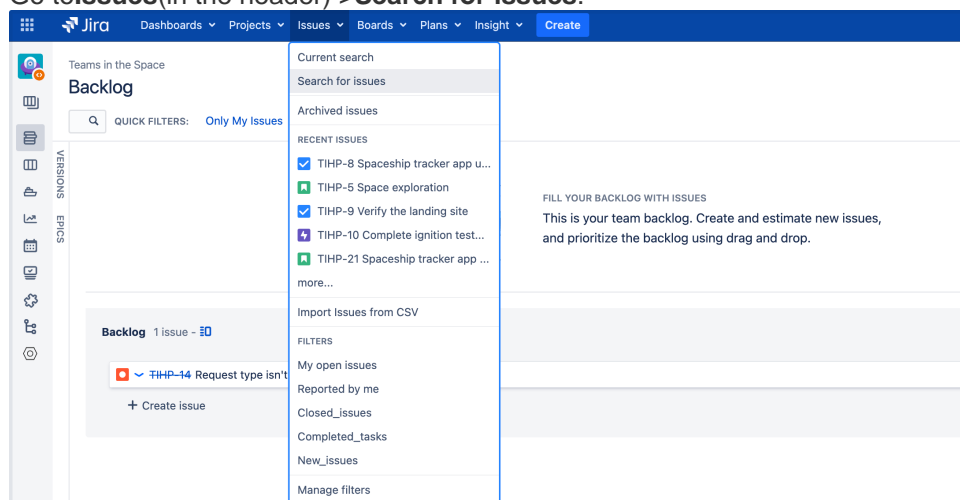
- If there are existing search criteria, select the **New search** button to reset the search criteria.

131

- If the basic search is shown instead of the advanced search, select **Advanced**(next to the **Sear ch** button).

> ⓘ 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 tha, 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 dont 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 forfunctionparameters.
   - Auto-complete suggestions are not offered for all fields. Check thefields 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.

  > ⓘ 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 assigne=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"

> (i) If youre 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 dont 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.

135

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

Fields	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.	To view a detailed information about fields and how to use them for advanced searching, check out Fields reference page.

To view a detailed information about fields and how to use them for advanced searching, check out Fields reference page.

- 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

Operators	An operator in JQL is one or more symbols or words that compare the value of afieldon its left with one or more values (orfunctions) on its right, such that only true results are retrieved by the clause. Some operators may use theNOTkeyword.	To view a detailed information about operators and how to use them for advanced searching, check out Operators reference 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
Keywords	A keyword in JQL is a word or phrase that does (or is) 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.	To view a detailed information about keywords and how to use them for advanced searching, check out Keywords reference page. • AND • OR • NOT • EMPTY • NULL • ORDER BY

| **Functions** | A function in JQL appears as a word followed by parentheses, which may contain one or more explicit values or Jira fields.

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. | To view a detailed information about functions and how to use them for advanced searching, check out Functions reference page.

- `approved()`
- `approver()`
- `cascadeOption()`
- `closedSprints()`
- `componentsLeadByUser()`
- `currentLogin()`
- `currentUser()`
- `earliestUnreleasedVersion()`
- `endOfDay()`
- `endOfMonth()`
- `endOfWeek()`
- `endOfYear()`
- `issueHistory()`
- `issuesWithRemoteLinksByGlobalId()`
- `lastLogin()`
- `latestReleasedVersion()`
- `linkedIssues()`
- `membersOf()`
- `myApproval()`
- `myPending()`
- `now()`
- `openSprints()`
- `pending()`
- `pendingBy()`
- `projectsLeadByUser()`
- `projectsWhereUserHasPermission()`
- `projectsWhereUserHasRole()`
- `releasedVersions()`
- `standardIssueTypes()`
- `startOfDay()`
- `startOfMonth()`
- `startOfWeek()`
- `startOfYear()`
- `subtaskIssueTypes()`
- `unreleasedVersions()`
- `votedIssues()`
- `watchedIssues()` |

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.
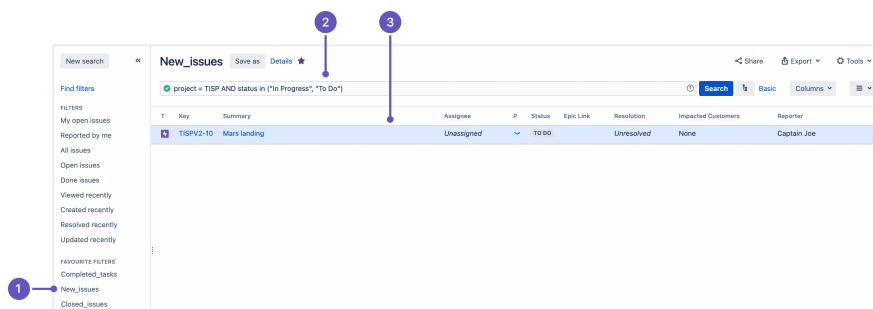
1. A **search saved as a filter**, which returns issues based on the criteria specified in a JQL query.
2. **JQL query** that specifies search criteria.
3. **Issues** that match the search criteria.

If you want to delete a saved search, see Deleting a filter.

Notes

To find out the version of Lucene Jira Software is using, go to `/Installation-directory/atlassian-jira/WEB-INF/lib`and locate the Lucene jar files. The Lucene version number will be part of the filename.

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	`affectedVersion`
Field Type	VERSION
Auto-complete	Yes

Supported operators	`= , != , > , >= , < , <= , ~ , !~`
	`IS, IS NOT, IN, NOT IN`
	
	• *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	`WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED`
Supported functions	When used with the**IN**and**NOT IN**operators, this field supports: • releasedVersions() • latestReleasedVersion() • unreleasedVersions() • earliestUnreleasedVersion()

Examples	• Find issues with an AffectedVersion of 3.14:
	`affectedVersion = "3.14"`
	
	Note that full-stops are reserved characters and need to be surrounded by quote-marks.
	• Find issues with an AffectedVersion of "Big Ted":
	`affectedVersion = "Big Ted"`
	• Find issues with an AffectedVersion ID of 10350:
	`affectedVersion = 10350`

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	`approvals`
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	- approved() - approver() - myApproval() - myPending() - pending() - pendingBy()
Examples	- Find issues that require or required approval by John Smith: `approval = approver(jsmith)` - Find issues that require approval by John Smith: `approval = pendingBy(jsmith)` - Find issues that require approval by the current user: `approval = myPending()` - Find all issues that require approval: `approval = pending()`

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	`assignee`
Alias	`cf[CustomFieldID]`
Field Type	USER
Auto-complete	Yes

3

Supported operators	= , != IS, IS NOT, IN, NOT IN, WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED *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.*
Unsupported operators	~ , !~ , > , >= , < , <=
Supported functions	When used with the**IN**and**NOT IN**operators, this field supports: • membersOf() When used with the**EQUALS**and**NOT EQUALS**operators, this field supports: • currentUser()

Examples	• 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"`
	
	 Note that full-stops and "@" symbols are reserved characters and need to be surrounded by quote-marks.

^ 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

Created in 2023 by Atlassian. Licensed under a Creative Commons Attribution 2.5 Australia License.

Examples	- Search for issues that have attachments: `attachments IS NOT EMPTY` - Search for issues that do not have attachments: `attachments IS EMPTY`

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	Find issues that belong to projects in the "Alphabet Projects" Category: `category = "Alphabet Projects"`

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	

	• 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: • componentsLeadByUser()

Examples	• 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: • currentLogin() • lastLogin() • now() • startOfDay() • startOfWeek() • startOfMonth() • startOfYear() • endOfDay() • endOfWeek() • endOfMonth() • endOfYear()

Examples	• 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

146

Search for issues that were created by a particular user.You can search by the user's full name, ID, or email address.

Syntax	creator
Field Type	USER
Auto-complete	Yes
Supported operators	**= , !=** **IS , IS NOT , IN , NOT IN, WAS, WAS IN, WAS NOT, WAS NOT IN**
Unsupported operators	**~ , !~ , > , >= , < , <=** **CHANGED**
Supported functions	When used with the**IN**and**NOT IN**operators, this field supports: • membersOf() When used with the**EQUALS**and**NOT EQUALS**operators, this field supports: • currentUser()

Examples	• Search for issues that were created by Jill Jones:
	`creator = "Jill Jones"`
	`or`
	`creator = "jjones"`
	• Search for issues that were created by the user with email address "bob@mycompany.com":
	`creator = "bob@mycompany.com"`
	
	*(Note that full-stops and "@" symbols are reserved*characters*, so the email address needs to be surrounded by quote-marks.)*

^ 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 fieldto 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	CustomFieldName
Alias	cf[CustomFieldID]
Field Type	*Depends on the custom field's configuration* *Note, Jira text-search syntaxcan be used with custom fields of type 'Text'.*

Auto-complete	Yes, for custom fields of type picker, group picker, select, checkbox and radio button fields
Supported operators	*Different types of custom field support differentoperators.*
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	*Different types of custom fields support differentfunctions.*
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: • currentLogin() • lastLogin() • now() • startOfDay() • startOfWeek() • startOfMonth() • startOfYear() • endOfDay() • endOfWeek() • endOfMonth() • endOfYear()

Supported functions: version picker fields	Version picker fields: When used with the **IN** and **NOT IN** operators, this field supports: • releasedVersions() • latestReleasedVersion() • unreleasedVersions() • earliestUnreleasedVersion()

Examples	• 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.

Syntax	`"Customer Request Type"`
Field Type	Custom field
Auto-complete	Yes

Supported operators	`= , !=`
	
	`IN , NOT IN`

Unsupported operators	`~ , !~ , > , >= , < , <=`
	`IS , IS NOT, WAS, WAS IN, WAS NOT, WAS NOT IN , CHANGED`
	
	ⓘ 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.

Supported functions	None

Examples	

	• 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 **Simpl eDesk 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	• 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: • currentLogin() • lastLogin() • now() • startOfDay() • startOfWeek() • startOfMonth() • startOfYear() • endOfDay() • endOfWeek() • endOfMonth() • endOfYear()

Examples	• 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 syntaxcan 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	- 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\""`

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	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** or **NOT IN** operators, `epic link` supports: - issueHistory() - linkedIssues() - votedIssues() - watchedIssues()

Examples	Find issues that belong to epic "Jupiter", where "Jupiter has the issue key ANERDS-31:
	`"epic link"= ANERDS- 31`
	`or`
	`"epic link"= Jupiter`

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	`filter`
Aliases	`request` , `savedFilter` , `searchRequest`
Field Type	Filter
Auto-complete	Yes
Supported operators	`= , !=` `IN , NOT IN`
Unsupported operators	`~ , !~ , > , >= , < , <=` `IS , IS NOT, WAS, WAS IN, WAS NOT, WAS NOT IN , CHANGED`
Supported functions	None

Examples	Search the results of the filter "My Saved Filter" (which has an ID of 12000) for issues assigned to the user jsmith:
	`filter = "My Saved Filter"and assignee = jsmith`
	or
	`filter = 12000and assignee = jsmith`

^ 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	`fixVersion`
Field Type	VERSION
Auto-complete	Yes

Supported operators	`= , != , > , >= , < , <= , ~ , !~`
	`IS , IS NOT, IN , NOT IN, WAS, WAS IN, WAS NOT, WAS NOT IN , CHANGED`
	    The comparison operators (e.g. ">") use theversion orderthat 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	When used with the**IN**and**NOT IN**operators, this field supports: • releasedVersions() • latestReleasedVersion() • unreleasedVersions() • earliestUnreleasedVersion()

Examples	• Find issues with a Fix Version of 3.14 or 4.2:
	`fixVersion in ( "3.14" , "4.2" )`
	
	(Note that full-stops are reserved characters, so they need to be surrounded by quote-marks.)
	• Find issues with a Fix Version of "Little Ted":
	`fixVersion = "Little Ted"`
	• Find issues with a Fix Version ID of 10001:
	`fixVersion = 10001`

^ 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	`issueKey`
Aliases	`id , issue , key`
Field Type	ISSUE
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, `issueKey` supports: • issueHistory() • linkedIssues() • updatedBy() • votedIssues() • watchedIssues()

Examples	• Find the issue with the key "ABC-123":
	`issueKey = ABC-123`
	• Find several issues with the known keys "SCRUM-25" and "SCRUM-12"
	`issuekey in (SCRUM-25, SCRUM-12)`

^ 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	• Find issues with a link type of "blocks":
	`issueLinkType = blocks`
	• Find issues with an issue type of "duplicates" or "is duplicated by":
	`issueLinkType in (duplicates,"is duplicated by")`
	• Find issues with link types other than clones:
	`issueLinkType != clones`
	  ℹ 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.
	`issueLinkType = "is blocked by"`
	`issueLinkType != "is blocked by"`

^ 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` *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.*
Unsupported operators	`~ , !~ , , > , >= , < , <=` `WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED`

Supported functions	None
Examples	- 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: - currentLogin() - lastLogin() - now() - startOfDay() - startOfWeek() - startOfMonth() - startOfYear() - endOfDay() - endOfWeek() - endOfMonth() - endOfYear()

Examples	

	• 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	• 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	- 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	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.

158

Syntax	`priority`
Field Type	PRIORITY
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	- Find issues with a priority of "High": `priority = High` - Find issues with a priority ID of 10000: `priority = 10000`

^ 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	`project`
Field Type	PROJECT
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, `project` supports: - projectsLeadByUser() - projectsWhereUserHasPermission() - projectsWhereUserHasRole()
Examples	- Find issues that belong to the Project that has the name "ABC Project": `project = "ABC Project"` - Find issues that belong to the project that has the key "ABC": `project = "ABC"` - Find issues that belong to the project that has the ID "1234": `project = 1234`

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	`remainingEstimate`
Alias	`timeEstimate`
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	• Find issues with a remaining estimate of more than 4 hours: remainingEstimate > 4h

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	`reporter`
Field Type	USER
Auto-complete	Yes
Supported operators	**= , !=** **IS , IS NOT, IN , NOT IN , WAS, WAS IN, WAS NOT, WAS NOT IN , CHANGED**
Unsupported operators	**~ , !~ , > , >= , < , <=**
Supported functions	When used with the**IN** and**NOT IN** operators, this field supports: • membersOf() When used with the**EQUALS**and**NOT EQUALS**operators, this field supports: • currentUser()

Examples	• Search for issues that were reported by Jill Jones:
	`reporter = "Jill Jones"`
	or
	`reporter = jjones`
	• Search for issues that were reported by the user with email address " bob@myc ompany.com ":
	`reporter = "bob@mycompany.com"`
	
	(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	`request-channel-type`
Field Type	TEXT
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: • `email`: requests submitted via email • `Jira`: requests created using Jira • `portal`: requests created using a Service Desk portal • `api`: requests created using a REST API

Examples	• Find issues where the request channel was email:
	`request-channel-type = email`
	• Find issues where the request channel was something other than a service desk portal:
	
	`request-channel-type != portal`

^ top of page

Request last activity time

Only applicable if Jira Service Management is installed and licensed.

Search for issues thatwere last actedon 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: - currentLogin() - lastLogin() - now() - startOfDay() - startOfWeek() - startOfMonth() - startOfYear() - endOfDay() - endOfWeek() - endOfMonth() - endOfYear()
Examples	- 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"`

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	- 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

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: • currentLogin() • lastLogin() • now() • startOfDay() • startOfWeek() • startOfMonth() • startOfYear() • endOfDay() • endOfWeek() • endOfMonth() • endOfYear()

Examples	• Find all issues that were resolved before 31st December 2010:
	`resolved <= "2010/12/31"`
	• Find all issues that were resolved before 2.00pm on 31st December 2010:
	`resolved < "2010/12/31 14:00"`
	• Find all issues that were resolved on or before 31st December 2010:
	`resolved <= "2011/01/01"`
	• Find issues that were resolved in January 2011:
	`resolved > "2011/01/01" and resolved < "2011/02/01"`
	• Find issues that were resolved on 15 January 2011:
	`resolved > "2011/01/15" and resolved < "2011/01/16"`
	• Find issues that were resolved in the last hour:
	resolved > -1h

^ 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	`~ , !~` `IS , IS NOT , IN , NOT IN , WAS, WAS IN, WAS NOT, WAS NOT IN , CHANGED`
Supported functions	• breached() • completed() • elapsed() • everBreached() • paused() • remaining() • running() • withinCalendarHours()

Examples	• 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	`= , !=` `IS , IS NOT, IN , NOT IN`
Unsupported operators	`~ , !~ , > , >= , < , <=` `WAS, WAS IN, WAS NOT, WAS NOT IN , CHANGED`
Supported functions	• openSprints() • closedSprints()

Examples	

	• Find issues that belong to sprint 999:
	`sprint = 999`
	• Find issues that belong to sprint "February 1":
	`sprint = "February 1"`
	• Find issues that belong to either "February 1", "February 2" or "February 3":
	`sprint in ( "February 1" , "February 2" , "February 3" )`
	• Find issues that are assigned to a sprint:
	sprint is not empty

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	`status`
Field Type	STATUS
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	• Find issues with a status of "Open": status = Open • Find issues with a status ID of 1: `status = 1` • Find issues that currently have, or previously had, a status of "Open": status WAS Open

Status category

Status category is a system field for grouping issue statuses. Each issue status in Jira can belong to one of the three status categories: **To Do**, **In Progress**, or **Done**. You can't add or remove status categories.

These status categories represent and generalize the three main stages of an ideal issue workflow. Each issue goes from the stage where the work on it hasn't started yet, through the stage when you're working on it, to the stage when the work on has been completed.

These stages can have multiple statuses that you set for your custom workflow. For example, the custom statuses "In development" and "In review" can belong to the single status category **In Progress**, because they represent the stage where you're developing and reviewing a feature described in the issue.

Syntax	`statusCategory`
Aliases	`New, Indeterminate,Complete`
Field Type	STATUS
Auto-complete	Yes
Supported operators	`= , !=, IN , NOT IN`

Unsupported operators	`~ , !~ , > , >= , < , <=`
	
	`IS , IS NOT, WAS, WAS IN, WAS NOT, WAS NOT IN , CHANGED`

Supported functions	None
Examples	- Find issues with the status category "To Do": `statusCategory = "To Do"` - Find issues with the status category ID `3` where `3` stands for closed issues: `statusCategory =3` - Find all issues that are currently in progress: `statusCategory not in ("To Do", "Done")`

^ top of page

Summary

Search for issues where the summary contains particular text. Jira text-search syntax can be used.

Syntax	`summary`
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	• 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\""`

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 theCONTAINSoperator ("~").
- Jira text-search syntaxcan 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	• Find issues where a text field matches the word "Fred":
	`text ~ "Fred"`
	`or`
	`text ~ Fred`
	• Find all issues where a text field contains the exact phrase "full screen":
	`text ~ "\"full screen\""`

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	`timeSpent`
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	• Find issues where the time spent is more than 5 days: `timeSpent > 5d`

^ 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 bytypeID thantypename. 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	`type`
Alias	`issueType`
Field Type	ISSUE_TYPE
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	• Find issues with an issue type of "Bug":
	`type = Bug`
	• Find issues with an issue typeof "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	`updatedDate`
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: • currentLogin() • lastLogin() • now() • startOfDay() • startOfWeek() • startOfMonth() • startOfYear() • endOfDay() • endOfWeek() • endOfMonth() • endOfYear()

Examples	- Find issues that were last updated before 12th December 2010: `updated < "2010/12/12"` - Find issues that were last updated on or before 12th December 2010: `updated < "2010/12/13"` - Find all issues that were last updated before 2.00pm on 31st December 2010: `updated < "2010/12/31 14:00"` - Find issues that were last updated more than two weeks ago: `updated < "-2w"` - Find issues that were last updated on 15 January 2011: `updated > "2011/01/15" and updated < "2011/01/16"` - Find issues that were last updated in January 2011: `updated > "2011/01/01" and updated < "2011/02/01"` - Find all issues updated since January 1, 2020: `updated >= "2020/01/01"`

^ 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.

Syntax	`voter`
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: - membersOf() When used with the **EQUALS** and **NOT EQUALS** operators, this field supports: - currentUser()
Examples	- Search for issues that you have voted for: `voter = currentUser()` - Search for issues that the user "jsmith" has voted for: `voter = "jsmith"` - Search for issues for which a member of the group "Jira-administrators" has voted: `voter in membersOf("Jira-administrators")`

^ top of page

Votes

Search for issues with a specified number of votes.

Syntax	`votes`
Field Type	NUMBER
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	Find all issues that have 12 or more votes: `votes >= 12`

^ 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	`watcher`
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: - membersOf() When used with the **EQUALS** and **NOT EQUALS** operators, this field supports: - currentUser()
Examples	- 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")`

172

Watchers

Search for issues with a specified number of watchers.

Syntax	`watchers`
Field Type	NUMBER
Auto-complete	No
Supported operators	`= , != , > , >= , < , <=` `IN , NOT IN`
Unsupported operators	`~ , !~` `IS , IS NOT , WAS, WAS IN, WAS NOT, WAS NOT IN , CHANGED`
Supported functions	When used with the **IN** and **NOT IN** operators, this field supports: • membersOf() When used with the **EQUALS** and **NOT EQUALS** operators, this field supports: • currentUser()
Examples	• Find all issues that are being watched by more than 3 people: `watchers > 3`

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: • membersOf() When used with the **EQUALS** and **NOT EQUALS** operators, this field supports: • currentUser()

Examples	• 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	• 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: • currentLogin() • lastLogin() • now() • startOfDay() • startOfWeek() • startOfMonth() • startOfYear() • endOfDay() • endOfWeek() • endOfMonth() • endOfYear()

Examples	

	• Find issues that have comments in work log entries created before midnight 00:00 12th December 2010:
	`worklogDate < "2010/12/12"`
	• Find issues that have comments in work log entries createdon or before 12th December 2010 (but not 13th December 2010):
	`worklogDate <= "2010/12/13"`
	• Find issues that have comments in work log entries createdon 12th December 2010 before 2:00pm:
	`worklogDate > "2010/12/12"and worklogDate < "2010/12/12 14:00"`
	• Find issues that have comments in work log entries createdless than one day ago:
	`worklogDate > "-1d"`
	• Find issues that have comments in work log entries created in January 2011:
	`worklogDate > "2011/01/01"and worklogDate < "2011/02/01"`
	• Find issues that have comments in work log entries created on 15 January 2011:
	`worklogDate > "2011/01/15"and worklogDate < "2011/01/16"`

^ 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.Workratiois calculated as follows:**workRatio = (timeSpent / originalEstimate) x 100**

Syntax	`workRatio`
Field Type	NUMBER
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	• Find issues on which more than 75% of the original estimate has been spent: `workRatio > 75`

^ top of page

Advanced searching - keywords reference

This page describes information about keywords that are used for advanced searching.

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 IN 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.

178

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 a DueDate, sorted by CreationDate:

```
duedate = empty order by created
```

- Find all issues without a DueDate, sorted by CreationDate, then by Priority (highest to lowest):

```
duedate = empty order by created, priority desc
```

- Find all issues without a DueDate, sorted by CreationDate, then by Priority (lowest to highest):

```
duedate = empty order by created, priority asc
```

> (i) 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).

179

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
  ```

- 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.

> ⓘ 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.

> ⓘ 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.

> ⓘ 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"
```

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.

> ⓘ 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")
```

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.

> ⓘ 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

> ⓘ The IN and NOT IN operators allow using up to 3000 operands.
>
> When the number of operands exceeds the allowed limit, the GET search fails because of the HTTP 400 error on Tomcat. In this case, we recommend using the `/search` resource through the POST method.

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 in `text ~ "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\""
```

> ✅ With this query, Jira will find issues where the summaries contain both the exact phrase "full screen" and any other phrase that includes the exact word combination "full screen". For example:
>
> - **"full screen"**
> - create **"full screen"**
> - **"full screen"** editing mode

- 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 doesn't match 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:

185

- Affected version
- Fix version
- Custom fields that use the Version Picker

> ✅ The JQL field "text", as in `text !~ "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"
```

  This query will return all issues where the value in the **Fix Version** field isn't 9.0, but it won't return issues where the **Fix Version** field is empty. To find issues where this field is empty or contains any other value except for 9.0, use the following query:

```
fixVersion !~ "9.0" OR fixVersion is empty
```

- 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 EMPTYorNULL 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, seethe 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"`

TheWASoperator 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, andStatus.

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".

> ⓘ 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, andStatus.

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"`

> ⓘ 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, andStatus.

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

> ⓘ The WAS IN and WAS NOT IN operators allow using up to 3000 operands.
>
> When the number of operands exceeds the allowed limit, the GET search fails because of the HTTP 400 error on Tomcat. In this case, we recommend using the `/search` resource through the POST method.

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 systemat 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, andStatus.

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"`

> (i) The operator can be used only with the following fields:Assignee,Fix Version,Priority,Reporter,Resolution, andStatus.

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 appscan add additional functions to the advanced issue search. For example, theScriptRunnerforJIRAapp extends JQL with such functions as `myProjects()` 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	`approved()`
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	- Find all issues that are approved: `Approvals = approved()` - Find all issues that have been approved by you or are pending your approval: `Approvals = myApproval() OR Approvals = myPending()`

^ top of page

On this page:

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	`approver(user,user)`
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	- Find issues that require or required approval by John Smith: `approval = approver(jsmith)` - Find issues that require or required approval by John Smith or Sarah Khan: `approval = approver(jsmith,skhan)`

^ 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	`breached()`
Supported fields	
Supported operators	**= , !=**
Unsupported operators	**~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED**
Examples	Find issues where Time to First Response was breached: `"Time to First Response" = breached()`

^ 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	`cascadeOption(parentOption)` `cascadeOption(parentOption,childOption)`
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	- Find issues where a custom field ("Location") has the value "USA" for the first tier and "New York" for the second tier: `location in cascadeOption("USA","New York")` - 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: `location in cascadeOption( "USA" )` - Find issues where a custom field ("Location") has the value "USA" for the first tier and no value for the second tier: `location in cascadeOption( "USA" ,none)` - Find issues where a custom field ("Location") has no value for the first tier and no value for the second tier: `location in cascadeOption(none)` - Find issues where a custom field ("Referrer") has the value "none" for the first tier and "none" for the second tier: `referrer in cascadeOption( "\"none\"" , "\"none\"" )` - Find issues where a custom field ("Referrer") has the value "none" for the first tier and no value for the second tier: `referrer in cascadeOption( "\"none\"" ,none)`

^ 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	`closedSprints()`
Supported fields	Sprint
Supported operators	`IN , NOT IN`
Unsupported operators	`= , != , ~ , !~ , > , >= , < , <=` `IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`
Examples	Find all issues that are assigned to a completedsprint: `sprint in closedSprints()`

^ 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	`completed()`
Supported fields	
Supported operators	`= , !=`
Unsupported operators	`= , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`
Examples	• Find issues where Time to First Response has completed at least one cycle: `"Time to First Response" = completed()`

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	`componentsLeadByUser()` `componentsLeadByUser(username)`
Supported fields	Component
Supported operators	`IN , NOT IN`
Unsupported operators	`= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`

Examples	• Find open issues in components that are led by you:
	`component in componentsLeadByUser() AND status = Open`
	• Find open issues in components that are led by Bill:
	`component in componentsLeadByUser(bill) AND status = Open`

currentLogin()

Perform searches based on the time at which the current user's session began. See also lastLogin().

Syntax	`currentLogin()`
Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time
Supported operators	**= , != , > , >= , < , <=** **WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*** **Only in predicate*
Unsupported operators	**~ , !~ , IS , IS NOT , IN , NOT IN**
Examples	Find issues that have been created during my current session: `created > currentLogin()`

^ 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	`currentUser()`
Supported fields	Assignee, Reporter, Voter, Watcher, custom fields of type User
Supported operators	**= , !=**
Unsupported operators	**~ , !~ , > , >= , < , <= , IS , IS NOT , IN , NOT IN , WAS ,** **WAS IN , WAS NOT , WAS NOT IN , CHANGED**
Examples	- Find issues that are assigned to me: `assignee = currentUser()` - Find issues that were reported to me but are not assigned to me: `reporter = currentUser() AND (assignee != currentUser() OR assignee is EMPTY)`

^ 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	`earliestUnreleasedVersion(project)`

Supported fields	AffectedVersion, FixVersion, custom fields of type Version
Supported operators	`= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`
Unsupported operators	`IN , NOT IN`
Examples	- Find issues whose FixVersion is the earliest unreleased version of the ABC project: `fixVersion = earliestUnreleasedVersion(ABC)` - Find issues that relate to the earlist unreleased version of the ABC project: `affectedVersion = earliestUnreleasedVersion(ABC) or fixVersion = earliestUnreleasedVersion(ABC)`

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	Find issues that have been waiting for a first response for more than 1 hour: `"Time to First Response" > elapsed("1h")`

endOfDay()

Perform searches based on the end of the current day. See also endOfWeek(), endOfMonth(), and endOfYear(), startOfDay(), startOfWeek(), startOfMonth(), and startOfYear().

Syntax	`endOfDay()`
	
	 `endOfDay("inc")`
	
	 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. endOfDay("+1") is the same as endOfDay("+1d"). If the plus/minus (+/-) sign is omitted, plus is assumed.

Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time

Supported operators	= , != , > , >= , < , <= WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED* **Only in predicate*
Unsupported operators	~ , !~ , IS , IS NOT , IN , NOT IN
Examples	- Find issues due by the end of today: `due < endOfDay()` - Find issues due by the end of tomorrow: `due < endOfDay("+1")`

endOfMonth()

Perform searches based on the end of the current month. See also endOfDay(), endOfWeek(), endOfYear(), startOfDay(), startOfWeek(), startOfMonth(), and startOfYear().

Syntax	`endOfMonth()`
	
	`endOfMonth("inc")`
	
	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.*

Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time
Supported operators	= , != , > , >= , < , <= WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED* **Only in predicate*
Unsupported operators	~ , !~ , IS , IS NOT , IN , NOT IN
Examples	- 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")`

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	`endOfWeek()` `endOfWeek("inc")` *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. endOfWeek("+1") is the same as endOfWeek("+1w"). If the plus/minus (+/-) sign is omitted, plus is assumed.*
Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time

Supported operators	`= , != , > , >= , < , <=`
	`WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*`
	`*Only in predicate`

Unsupported operators	`~ , !~ , IS , IS NOT , IN , NOT IN`

Examples	• Find issues due by the end of this week:
	`due < endOfWeek()`
	• Find issues due by the end of next week:
	`due < endOfWeek("+1")`

^ 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	`endOfYear()` `endOfYear("inc")` *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. endOfYear("+1") is the same as endOfYear("+1y"). If the plus/minus (+/-) sign is omitted, plus is assumed.*
Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time

Supported operators	`= , != , > , >= , < , <=`
	`WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*`
	`*Only in predicate`

Unsupported operators	`~ , !~ , IS , IS NOT , IN , NOT IN`

Examples	• Find issues due by the end of this year:
	`due < endOfYear()`
	• Find issues due by the end of March next year:
	`due < endOfYear("+3M")`

^ top of page

everbreached()

Only applicable if Jira Service Management is installed and licensed.

Returns issues that have missed one of their goals.

Syntax	`elapsed()`
Supported fields	
Supported operators	`= , !=`
Unsupported operators	`~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`
Examples	• Find issues have missed their goal for Time to First Response: `"Time to First Response" = everbreached()`

^ 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	`futureSprints()`
Supported fields	Sprint
Supported operators	`IN , NOT IN`
Unsupported operators	`= , != , ~ , !~ , > , >= , < , <=` `IS , IS NOT, WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`
Examples	• Find all issues that are assigned to asprint that hasn't been started yet: `sprint in futureSprints()`

^ 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	`issueHistory()`
Supported fields	Issue
Supported operators	**IN , NOT IN**
Unsupported operators	**= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED**
Examples	• Find issues which I have recently viewed, that are assigned to me: `issue in issueHistory() AND assignee = currentUser()`

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	`issuesWithRemoteLinksByGlobalId()`
Supported fields	Issue
Supported operators	**IN , NOT IN**
Unsupported operators	**= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED**

Examples	• Find issues that are linked to remote links that have globalId "abc":
	  `issue in issuesWithRemoteLinksByGlobalId(abc)`
	• Find issues that are linked to remote links that have either globalId "abc" or "def":
	  `issue in issuesWithRemoteLinksByGlobalId(abc, def)`

lastLogin()

Perform searches based on the time at which the current user's previous session began. See also currentLogin().

Syntax	`lastLogin()`
Supported fields	Created. Due, Resolved, Updated, custom fields of type Date/Time

Supported operators	`= , != , > , >= , < , <=`
	`WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*`
	
	`*`*Only in predicate*

Unsupported operators	`~ , !~ ,IS , IS NOT , IN , NOT IN`
Examples	Find issues that have been created during my last session: `created > lastLogin()`

^ 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 alsoreleasedVersions().

Consider that the "latest" is determined by the ordering assigned to the versions, not by actual Version Due Dates.

Syntax	`latestReleasedVersion(project)`
Supported fields	AffectedVersion, FixVersion, custom fields of type Version
Supported operators	`= , !=`

Unsupported operators	`~ , !~ , > , >= , < , <=`
	
	`IS , IS NOT , IN , NOT IN ,WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`

Examples	- Find issues whose FixVersion is the latest released version of the ABC project: `fixVersion = latestReleasedVersion(ABC)` - Find issues that relate to the latest released version of the ABC project: `affectedVersion = latestReleasedVersion(ABC) or fixVersion = latestReleasedVersion(ABC)`

^ 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	`linkedIssues(issueKey)` `linkedIssues(issueKey,linkType)`
Supported fields	Issue

Supported operators	`IN , NOT IN`
Unsupported operators	`= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`

Examples	• 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	• 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	**~, != ,!~ ,>,>=,<,<=** **IS , IS NOT ,IN , NOT IN ,WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED**
Examples	• Find all issues that require or have required my approval `approval = myApproval()`

myPending()

Only applicable if Jira Service Management is installed and licensed.

Search for issues that require approval by the current user.

Syntax	`myPending()`
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	• Find all issues that require my approval `approval = myPending()`

now()

Perform searches based on the current time.

Syntax	`now()`
Supported fields	Created. Due, Resolved, Updated, custom fields of type Date/Time
Supported operators	**= , != , > , >= , < , <=** **WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*** **Only in predicate*
Unsupported operators	**~ , !~ ,IS , IS NOT , IN , NOT IN**
Examples	• Find issues that are overdue: `duedate < now() and status not in (closed, resolved)`

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	`openSprints()`
Supported fields	Sprint
Supported operators	`IN , NOT IN`

Unsupported operators	`= , != , ~ , !~ , > , >= , < , <=`
	`IS , IS NOT, WAS , WAS IN , WAS NOT , WAS NOT IN ,`
	`CHANGED`

Examples	• Find all issues that are assigned to asprint that has not yet been completed: `sprint in openSprints()`

^ top of page

outdated()

Only applicable if Jira Service Management (Server) is installed and licensed.

Returns issues whose SLAs are out of date because someone has changed the SLA in the settings. After the site reindexes and recalculates the SLAs, the function shouldn't return any issues. Use this function if a reindex is taking a long time or if you've deferred the reindex because you're making a lot of changes.

Syntax	`outdated()`
Supported fields	
Supported operators	`= , !=`
Unsupported operators	`~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS` `NOT , WAS NOT IN , CHANGED`
Examples	• Find issues where SLAs are out of date: `"Time to First Response"= outdated()`

^ top of page

paused()

Only applicable if Jira Service Management is installed and licensed.

Returns issues that have an SLA that is paused due to a condition.

To find issues that are paused because they are outside calendar hours, usewithincalendarhours().

Syntax	`paused()`
Supported fields	
Supported operators	**= , !=**
Unsupported operators	**~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED**
Examples	• Find issues where Time to First Response is paused: `"Time to First Response" = paused()`

pending()

Only applicable if Jira Service Management is installed and licensed.

Search for issues that require approval.

Syntax	`pending()`
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	• Find all issues that require approval: `approval = pending()`

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	`pendingBy(user1,user2)`
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	- Find issues that require approval by John Smith:
	`approval = pendingBy(jsmith)`
	- Find issues that require by John Smith or Sarah Khan:
	`approval = pendingBy(jsmith,skhan)`

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.

> (i) If you are not logged in to Jira, a user must be specified.

Syntax	`projectsLeadByUser()`
	
	`projectsLeadByUser(username)`

Supported fields	Project
Supported operators	IN , NOT IN
Unsupported operators	= , != , ~ , !~ , > , >= , < , <= IS , IS NOT, WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED

Examples	- Find open issues in projects that are led by you:
	`project in projectsLeadByUser() AND status = Open`
	- Find open issues in projects that are led by Bill:
	`project in projectsLeadByUser(bill) AND status = Open`

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 arenot 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	`projectsWhereUserHasPermission(permission)` For the`permission`parameter, you can specify any of the permissions described on .
Supported fields	Project
Supported operators	IN , NOT IN

Unsupported operators	`= , != , ~ , !~ , > , >= , < , <=` `IS , IS NOT, WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`
Examples	• Find open issues in projects where you have the "Resolve Issues" permission: `project in projectsWhereUserHasPermission("Resolve Issues") AND status = Open`

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	`projectsWhereUserHasRole(rolename)`
Supported fields	Project
Supported operators	`IN , NOT IN`
Unsupported operators	`= , != , ~ , !~ , > , >= , < , <=` `IS , IS NOT, WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`
Examples	• Find open issues in projects where you have the "Developers" role: `project in projectsWhereUserHasRole("Developers") AND status = Open`

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 alsolatestReleasedVersion().

Syntax	`releasedVersions()` `releasedVersions(project)`
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	• Find issues whose FixVersion is a released version of the ABC project:
	`fixVersion in releasedVersions(ABC)`
	• Find issues that relate to released versions of the ABC project:
	`(affectedVersion in releasedVersions(ABC)) or (fixVersion in releasedVersions(ABC))`

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	`remaining()`
Supported fields	
Supported operators	`= , !=, > , >= , < , <=`
Unsupported operators	`~ , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`
Examples	• Find issues that will breach Time to Resolution in the next two hours: `"Time to Resolution" < remaining("2h")`

running()

Only applicable if Jira Service Management is installed and licensed.

Returns issues that have an SLA that is running, regardless of the calendar.

To find issues that are running based on calendar hours, use withincalendarhours().

Syntax	`running()`
Supported fields	
Supported operators	`= , !=`
Unsupported operators	`~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`
Examples	• Find issues where Time to First Response is running: `"Time to First Response" = running()`

standardIssueTypes()

Perform searches based on "standard" Issue Types, that is, search for issues that are not sub-tasks.See also subtaskIssueTypes().

Syntax	`standardIssueTypes()`
Supported fields	Type
Supported operators	**IN , NOT IN**
Unsupported operators	**= , != , ~ , !~ , > , >= , < , <= IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED**
Examples	Find issues that are not subtasks (i.e. issues whose Issue Type is a standard issue type, not a subtask issue type): `issuetype in standardIssueTypes()`

^ top of page

startOfDay()

Perform searches based on the start of the current day. See alsostartOfWeek(),startOfMonth(), startOfYear(), endOfDay(),endOfWeek(),endOfMonth(),andendOfYear().

Syntax	`startOfDay()`
	
	`startOfDay("inc")`
	
	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. startOfDay("+1") is the same as startOfDay("+1d"). If the plus/minus (+/-) sign is omitted, plus is assumed.

Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time
Supported operators	**= , != , > , >= , < , <=** **WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*** **Only in predicate*
Unsupported operators	**~ , !~ , IS , IS NOT , IN , NOT IN**
Examples	- Find new issues created since the start of today: `created > startOfDay()` - Find new issues created since the start of yesterday: `created > startOfDay("-1")` - Find new issues created in the last three days: `created > startOfDay("-3d")`

^ 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	`startOfMonth()`
	
	`startOfMonth("inc")`
	
	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. startOfMonth("+1") is the same as startOfMonth("+1M"). If the plus/minus (+/-) sign is omitted, plus is assumed.

Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time

Supported operators	`= , != , > , >= , < , <=`
	`WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*`
	`*Only in predicate`

Unsupported operators	`~ , !~ , IS , IS NOT , IN , NOT IN`
Examples	- Find new issues created since the start of this month: `created > startOfMonth()` - Find new issues created since the start of last month: `created > startOfMonth("-1")` - Find new issues created since the 15th of this month: `created > startOfMonth("+14d")`

^ 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	`startOfWeek()`
	
	`startOfWeek("inc")`
	
	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. startOfWeek("+1") is the same as startOfWeek("+1w"). If the plus/minus (+/-) sign is omitted, plus is assumed.

Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time

Supported operators	`= , != , > , >= , < , <=`
	`WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*`
	`*Only in predicate`

Unsupported operators	`~ , !~ , IS , IS NOT , IN , NOT IN`

Examples	- Find new issues since the start of this week: `created > startOfWeek()` - Find new issues since the start of last week: `created > startOfWeek("-1")`

startOfYear()

Perform searches based on the start of the current year. See also startOfDay(), startOfWeek(), startOfMonth(), endOfDay(), endOfWeek(), endOfMonth(), and endOfYear.

Syntax	`startOfYear()`
	
	`startOfYear("inc")`
	
	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. endOfYear("+1") is the same as endOfYear("+1y"). If the plus/minus (+/-) sign is omitted, plus is assumed.

Supported fields	Created, Due, Resolved, Updated, custom fields of type Date/Time
Supported operators	**= , != , > , >= , < , <=** **WAS* , WAS IN* , WAS NOT* , WAS NOT IN* , CHANGED*** ***Only in predicate*
Unsupported operators	**~ , !~ , IS , IS NOT , IN , NOT IN**
Examples	- Find new issues since the start of this year: `created > startOfYear()` - Find new issues since the start of last year: `created > startOfYear("-1")`

subtaskIssueTypes()

Perform searches based on issues that are sub-tasks. See also standardIssueTypes().

Syntax	`subtaskIssueTypes()`
Supported fields	Type
Supported operators	**IN , NOT IN**
Unsupported operators	**= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED**

Examples	• Find issues that are subtasks (i.e. issues whose Issue Type is a subtask issue type): `issuetype in subtaskIssueTypes()`

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	`unreleasedVersions()` `unreleasedVersions(project)`
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	• Find issues whose FixVersion is an unreleased version of the ABC project: `fixVersion in unreleasedVersions(ABC)` • Find issues that relate to unreleased versions of the ABC project: `affectedVersion in unreleasedVersions(ABC)`

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 then a day, and will always round them up to 1 day.

Syntax	`updatedBy(user)` `updatedBy(user, dateFrom)` `updatedBy(user, dateFrom, dateTo)`
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	- Find issues that were updated by John Smith: `issuekey IN updatedBy(jsmith)` - Find issues that were updated by John Smith within the last 8 days: `issuekey IN updatedBy(jsmith, "-8d")` - Find issues updated between June and September 2018: `issuekey IN updatedBy(jsmith, "2018/06/01", "2018/08/31")` - 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: `issuekey IN updatedBy(jsmith, "-1h")`

^ 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	`votedIssues()`
Supported fields	Issue
Supported operators	`IN , NOT IN`
Unsupported operators	`= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`
Examples	Find issues that you have voted for: `issue in votedIssues()`

^ top of page

watchedIssues()

Perform searches based on issues that you are watching. Also, see theWatcher field.

ⓘ This function can only be used by logged-in users.

Syntax	`watchedIssues()`
Supported fields	Issue

Supported operators	`IN , NOT IN`
Unsupported operators	`= , != , ~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`
Examples	Find issues that you are watching: `issue in watchedIssues()`

^ top of page

withinCalendarHours()

Only applicable if Jira Service Management is installed and licensed.

Returns issues that have an SLA 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	`withinCalendarHours()`
Supported fields	
Supported operators	`= , !=`
Unsupported operators	`~ , !~ , > , >= , < , <= , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`
Examples	Find issues where Time to First Response is within calendar hours: `"Time to First Response" = withinCalendarHours()`

^ 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.

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 youre 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 arent 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.

> (i) 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 looks 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"
```

> (i) 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 arent stored in the index, which means you cant 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 (`[ ]`) wont 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 doesnt apply to Jira 8.x. If youd 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

Screenshot: Issue filter results in detail view
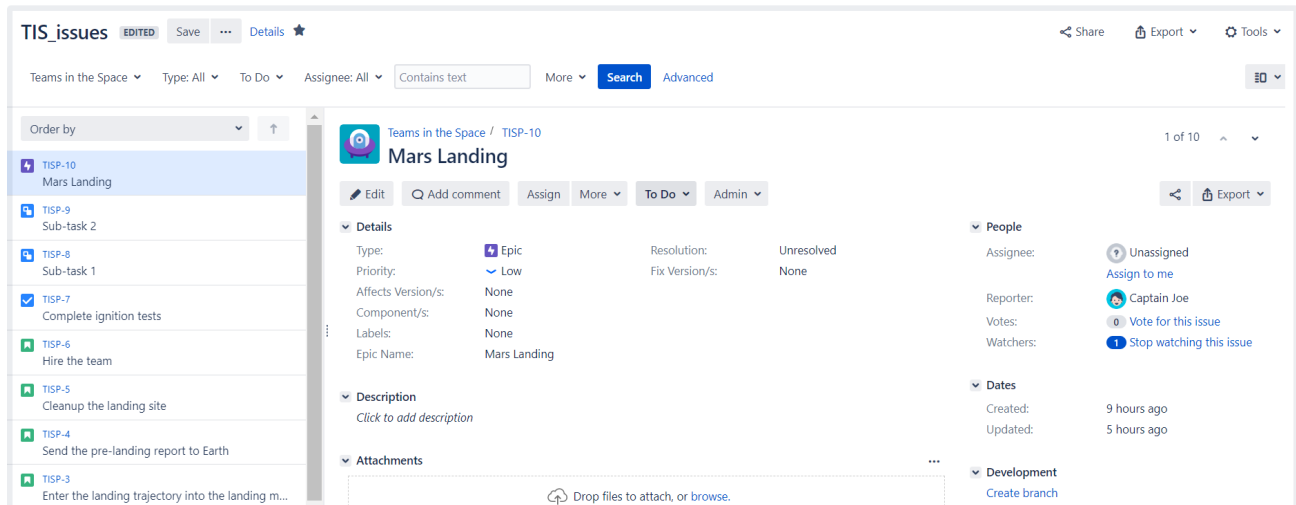
Saving a search as a filter

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.

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, eitherclone (aka copy) the shared filter, or ask your Jira administrator tochange 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.

> ⓘ 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.

The following screenshot provides an overview of the key features of the issue navigator.

Screenshot: Issue navigator (Detail view)
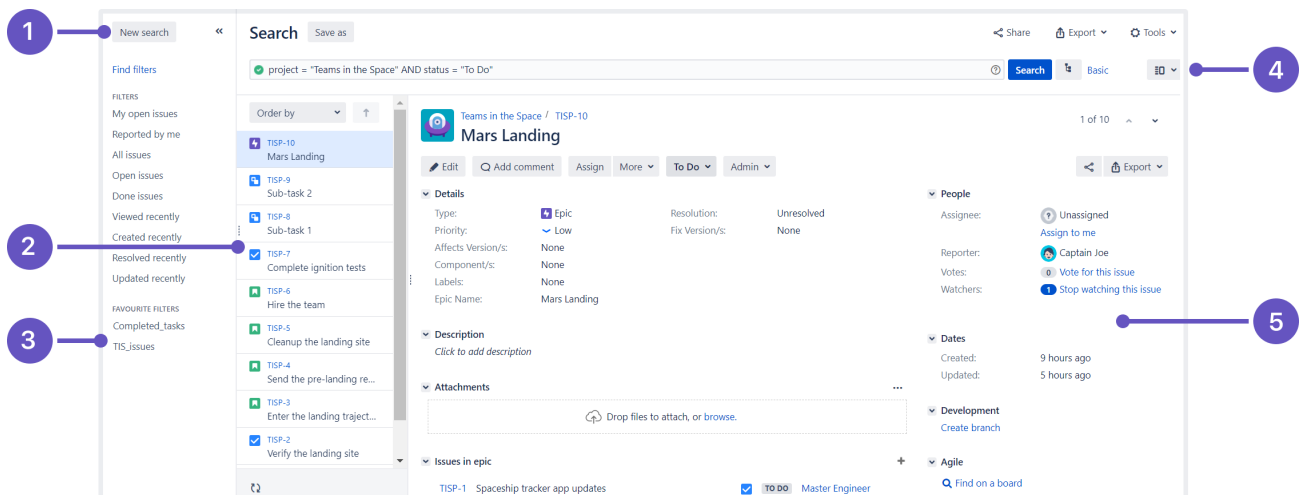
1. **Filter panel:**Click<<to collapse the filter panel so you can have more space in the detail view.
2. **Issue:**Select an issue from this panel to see the details in the detail view window.
3. **Filters:**Select a filter to see all the matching issues in the panel to the immediate right.
4. **Views:**Click 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

List view or Detail view	Click the **Change view** (☰ ˅) dropdown to switch between List view and Detail view for your search results. • **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).
Change the sort order	Click the column name. If you click the same column name more than once, the sort order will switch between ascending and descending. Note: • You cannot sort by the 'Images' column nor thesub-taskaggregate columns (i.e. all columns beginning with ''). • If you sort the search results for anadvanced search, an 'ORDER BY' clause will be added/updated for your JQL query to reflect the order of issues in your search results.
Columns show /hide and move	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: • **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. You can also modify any of these configurations. Make sure you have switched the desired configuration, then do the following: • 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. If you cannot find a column, please make sure that you haven't run in to any of the following restrictions: • You can only see columns for issue fields that have not beenhiddenand that you havepermissionsto see. • It is possible to add any of the existingcustom fieldsto 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 thatyou have a constant set of search results that you can work from when triaging issues.

View an issue	Click the key or summary of the issue. • If you are in List view, you will 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	To action an issue (e.g. edit it, transition it, log work on it, etc): • 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. You can also select an issue and action it via keyboard shortcuts in either views. *Tip: use the **'j'** and **'k'** keys to select the previous/next issue in the issue navigator.*

Sharing your search results

Click**Share**in the issue navigator toemail a link to asearch resultor 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 yourusername) 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, seeJira 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, seeReporting.

Exporting your search results

CSV	Click **Export** > **CSV (All fields)** or **Export** > **CSV (Current fields)**, and choose a delimiter to separate the values. The CSV file will contain a header row with every applicable issue field, comment, and attachment in your search result. • **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. 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. 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

HTML	Click **Export** > **HTML (All fields)** or **Export** > **HTML (Current fields)**. The HTML file will contain a header row with a value for every applicable issue field in your search result. • **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. 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.
XML	Click **Export** > **XML**. 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. 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.
Word	Click **Export** > **Word.** 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.

List of fields for field parameter (XML exports):

Value	Sample XML output	Notes
title	`<title>[TEST-4] This is a test</title>`	
link	`<link>https://extranet.atlassian.com:443/Jira/browse/TEST-4</link>`	This is a 'permalink' to the issue. For links between issues, see * `issuelinks` * (below).
project (or **pid**)	`<project id="10330" key="TST">Test</project>`	
description	`<description>This is a detailed description of the issue.<description>`	
environment	`<environment>Sydney network<environment>`	
key	`<key id="22574">TEST-4</key>`	

summary	`<summary>This is a test<summary>`	
type (or **issuetype**)	`<type id="3" iconUrl="https://extranet. atlassian.com:443/Jira/images/icons/task. gif">Task</type>`	
parent	`<parent id="22620">TEST-5</parent>`	Only relevant if the issue is a sub-task.
priority	`<priority id="4" iconUrl=" https://extranet.atlassian.com:443/Jira /images/icons/priority_minor.gif" >Minor<priority>`	
status	`<status id="5" iconUrl="https://extranet. atlassian.com:443/Jira/images/icons /status_resolved.gif">Resolved</status>`	
resolution	`<resolution id="1">Fixed</resolution>`	
labels	`<labels> <label>focus</label> <labels>`	
assignee	`<assignee username="jsmith">John Smith< /assignee>`	
reporter	`<assignee username="jsmith">John Smith< /assignee>`	
security	`<security id="10021">Private</security>`	Only relevant if a security level has been applied to the issue.
created	`<created>Mon, 1 Sep 2008 17:30:03 -0500 (CDT)>/created>`	
updated	`<updated>Mon, 1 Sep 2008 17:30:03 -0500 (CDT)>/updated>`	
resolved (or **resolutiondate**)	`<resolved>Mon, 1 Sep 2008 17:30:03 -0500 (CDT)>/resolved>`	

due (or **d uedate**)	`<due>Mon, 1 Sep 2008 17:30:03 -0500 (CDT) >/due>`	
version (or **versi ons**)	`<version>2.4.7</version>`	
fixfor (or **fixVe rsions**)	`<fixVersion>2.6</fixVersion>`	
compone nt (or **co mponents**)	`<component>Documentation</component>`	
votes	`<votes>1</votes>`	

comments (or **comme nt**)	`<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>`	

attachm ents (or **attachm ent**)	`<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>`
		Only available if your administrator has enabled attachments.

timeori ginales timate	`<timeoriginalestimate seconds="600">10 minutes</timeoriginalestimate>`	Only available if your administrator has time tracking enabled.
timeest imate	`<timeestimate seconds="300">5 minutes< /timeestimate>`	Only available if your administrator has time tracking enabled.
timespe nt	`<timespent seconds="300">5 minutes< /timespent>`	Only available if your administrator has time tracking enabled.
aggrega tetimeo riginal estimate	`<aggregatetimeoriginalestimate seconds=" 36000">10 hours< /aggregatetimeoriginalestimate>`	Aggregate time for the issue plus all of its sub-tasks. Only available if your administrator has time tracking enabled.

aggrega tetimee stimate	`<aggregatetimeremainingestimate seconds="18000">5 hours</aggregatetimeremainingestimate>`	Aggregate time for the issue plus all of its sub-tasks. Only available if your administrator has time tracking enabled.
aggrega tetimes pent	`<aggregatetimespent seconds="18000">5 hours</aggregatetimespent>`	Aggregate time for the issue plus all of its sub-tasks. Only available if your administrator has time tracking enabled.

timetra cking	`<timeoriginalestimate seconds="600">10 minutes</timeoriginalestimate>`
	`<timeestimate seconds="300">5 minutes</timeestimate>`
	`<timespent seconds="300">5 minutes</timespent>`
	`<aggregatetimeoriginalestimate seconds="36000">10 hours</aggregatetimeoriginalestimate>`
	`<aggregatetimeremainingestimate seconds="18000">5 hours</aggregatetimeremainingestimate>`
	`<aggregatetimespent seconds="18000">5 hours</aggregatetimespent>`
		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.

issueli nks	`<issuelinks>`
	`    <issuelinktype id="10020">`
	`        <name>Duplicate</name>`
	`        <inwardlinks description="is duplicated by">`
	`            <issuelink>`
	`                <issuekey id="22477">INTSYS-1009</issuekey>`
	`            </issuelink>`
	`        </inwardlinks>`
	`    </issuelinktype>`
	`</issuelinks>`	

subtasks (or subta sk)	`<subtasks>`
	`    <subtask id="22623">TEST-8</subtask>`
	`</subtasks>`	

customf ield_xx xxx	`<customfields>`
	`    <customfield id="customfield_10112" key="com.atlassian.Jira.plugin.system.customfieldtypes:select">`
	`        <customfieldname>Department</customfieldname>`
	`        <customfieldvalues>`
	`<customfieldvalue>Adminstration</customfieldvalue>`
	`        </customfieldvalues>`
	`    </customfield>`
	`</customfields>`
		"xxxxx" is the id of a given custom field e.g. this output is the result of specifying `&field=customfield_10112`

allcust om	```xml
	<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>
	``

Printable views

Printable	Click **Export** > **Printable**. 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.
Full content	Click **Export** > **Full content**. 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.

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	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.
	
	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**.
	
	More information:
	
	• 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:
	    • 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 `currentUser()` 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	Click **Export** > **RSS (Issues)** or **Export** > **RSS (Comments)**. The URL of the page that shows can be used in your feed reader. Tips: • 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.`http://mycompany.com/anypage?os_authType=basic`) 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, seeEditing 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 at8:15 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 theExamples sectionat 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 *within the same month,* 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.

Reporting

Jira Core provides a range of reports that show statistics for particular people, projects, versions, or information aboutissues.

The documentation in this section will help you configure and use the reports in Jira Core.

Search the topics in 'Reporting':

Generating a report

To generate a report:

1. Navigate to the desired project and click **Reports**.
2. Select a report from the list. See the 'Reports' section below for information about each report.

Reports

Chart	Purpose
Average Age Report	Shows the average age of unresolved issues for a project or filter. This helps you see whether your backlog is being kept up to date.
Created vs Resolved Issues Report	Maps created issues versus resolved issues over a period of time. This helps you understand whether your overall backlog is growing or shrinking. • Viewing the chart Areas in red show periods where more issues were created than resolved. Areas in green show periods where more were resolved than created.
Pie Chart Report	Shows a pie chart of issues for a project or filter grouped by a specified field. This helps you see the breakdown of a set of issues, at a glance. For example, you could create a chart to show issues grouped by Assignee for a particular version in a project (using a filter).
Recently Created Issues Report	Shows the number of issues created over a period of time for a project or filter, and how many were resolved. This helps you understand if your team is keeping up with incoming work. • Viewing the chart The green portion of the bar shows the created issues that are resolved. The red portion shows created but unresolved issues as yet.
Resolution Time Report	Shows the length of time taken to resolve a set of issues for a project or filter. This helps you identify trends and incidents that you can investigate further.
Single Level Group By Report	Shows issues grouped by a particular field for a filter. This helps you group search results by a field, and see the overall status of each group. For example, you could view the issues in a version of a project, grouped by Assignee.
Time Since Issues Report	For a date field and project or filter, maps the issues against the date that the field was set. This can help you track how many issues were created, updated, etc over a period of time.
Time	Showstime tracking informationon issues for a particularversionof a project.

Tracking Report *	The table in the report shows the issues within the version:

- There are four time tracking fields as follows:
  - **Original Estimate** - The original estimate of the total amount of time it would take to complete this issue.
  - **Estimated Time Remaining** - The current estimate of the remaining amount of time it would take to complete this issue.
  - **Time Spent** - The amount of time spent on the issue. This is the aggregate amount of time that has been logged against this issue.
  - **Accuracy** - The accuracy of the original estimate compared to the current estimate for the issue. It is the difference between the sum of the **Time Spent** and **Estimated Time Remaining** fields, and the **Original Estimate** field.
- If sub-tasks are enabled, the *"*column at the right of the field shows the aggregate time tracking information for each 'parent' issue (i.e. the sum of the issue's own values, plus those of its sub-tasks).
- The last line of the table shows the aggregate time tracking information for the whole version.

The report also includes two bar-graphs (above the table), which represent the aggregate time tracking information for the version:

- The first bar-graph **('Progress')** shows the percentage of completed issues (green) and incomplete issues (orange) in this version:

  Progress: 40%

- The second bar-graph **('Accuracy'** -blue) shows the accuracy of the original estimates.

The length of the **Accuracy** bar compared to the **Progress** bar indicates whether the issues in this version are ahead of or behind schedule. There are three cases:

1. *The issues are on schedule with the original estimate.*
   The **Accuracy** bar is completely blue and is the same length as the **Progress** bar above it.

   Progress: 40%
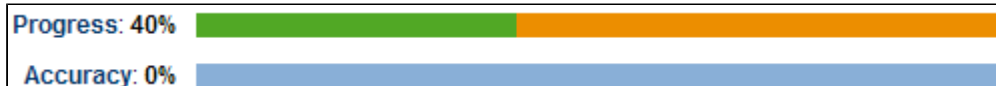
   Accuracy: 0%

2. *The issues are behind the original estimate (i.e. will take longer than originally estimated).*
   The **Progress** graph is longer than the **Accuracy** graph. The blue region represents the original estimated time, and the light-grey region is the amount of time by which issues are behind.

   Progress: 42%

   Accuracy: -4%

3. *The issues are ahead of the original estimate (i.e. will take less time than originally estimated).*
   The **Accuracy** graph is longer than the **Progress** graph. The blue bar represents the original estimated time, and the light-grey region represents the amount of time by which the original estimates were overestimated.

   Progress: 47%

   Accuracy: 8%
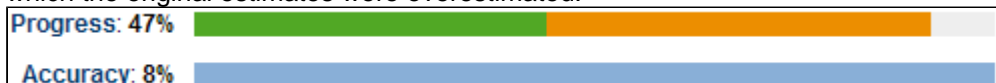
When generating the time tracking report, consider the following settings:

1. For **fix version**, choose the version on which you wish to report. The report will include all issues that belong to this version, that is, all issues whose**'Fix Version'**is this version.
2. For **sorting**, choose how the issues in the report will be sorted:
   - **Least completed issues first** shows issues with the highest**Estimated Time Remaining**first
   - **Most completed issues first** shows issues with the lowest**Estimated Time Remaining**first
3. For **issues**, choose which issues will be included in the report:
   - **All** includes all issues assigned to this version
   - **Incomplete issues only** excludes issues which are either completed (i.e. have an**Estimated Time Remaining**of zero), or are not time-tracked (i.e. do not have an**Original Estimate**).
     Note that issue status does not affect which issues are displayed.
4. For **sub-task inclusion** *(note: this will only appear ifsub-tasksare enabled)*, choose which sub-tasks will be included in the report, for all parent issues that belong to this version:
   - **Only include sub-tasks with the selected version** includes an issue's sub-tasks only if the sub-tasks belong to the same version as the issue
   - **Also include sub-tasks without a version set** includes an issue's sub-tasks if the sub-tasks belong to either the same version as the issue or to no version
   - **Include all sub-tasks** includes all of an issue's sub-tasks, regardless of whether the sub-tasks belong to the same version, some other version or no version.
     Note that sub-tasks which belong to this version, but whose parent issues do*not* belong to this version, will always be included in the report.

User Workload Report*	Shows how much work a user has been allocated, and how long it should take. For a specified user, you'll be able to see the number of unresolved issues assigned to the specified user, and the remaining workload, on a per-project basis.
Version Workload Report*	Shows how much outstanding work there is (per user and per issue) before a given version is complete. For the specified version, you'll be able to see a list of unresolved issues assigned to each user, each user's workload, and a summary of the total remaining workload for the version.
Workload Pie Chart Report*	Shows the relative workload for assignees of all issues for a project or filter.

**Only available if your JIRA administrator hasenabled time tracking.*

Reports available in Confluence

If you have connectedJira to Confluence, you can create the following reports in Confluence.

Chart	Purpose
Change Log	Displays a list of issues from Jira. This list can be static or dynamic, automatically updating as the status of your issues change in Jira.
Status Report	The Status Report displays the progress of a Jira project and fix version in pie charts by status, priority, component, and issue type.The Status Report uses the Jira Chart macro, and is dynamic.

Other reports

- Additional reports (e.g. Gantt Chart Report, Timesheet Report, JiraSQL Plugin) are available for download from the Atlassian Marketplace.
- Jira administrators can also create new reports with the app API see our Tutorial - Creating a Jira report . If you don't want to build an app yourself, Atlassian Experts are available for custom projects.
- Issue filters can be exported to Microsoft Excel, where they can be further manipulated into charts and reports. See Working with search results.

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 adefault dashboard, or you cancreatea personal dashboard and add gadgets to keep track of assignments and issues you're working on. Dashboards are designed to displaygadgetsthat 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 youruser profile,andchange dashboard permissions at any time in theManageDashboardspage.

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:
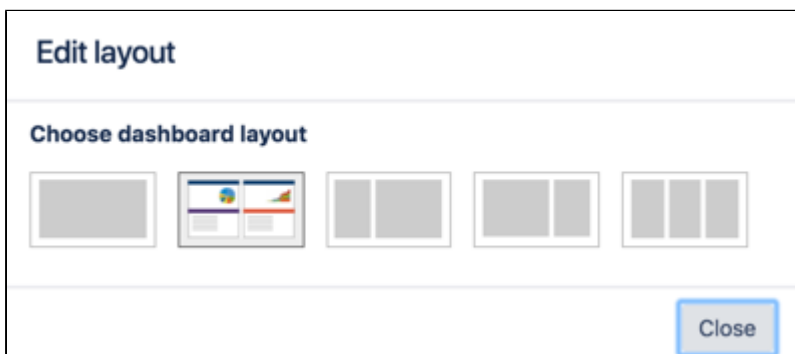
2. Choose your preferred layout.

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 staricon next to its name to add it to your favorite dashboards list. You can also add thedefault dashboardto your favorites list so it's easily available to you.
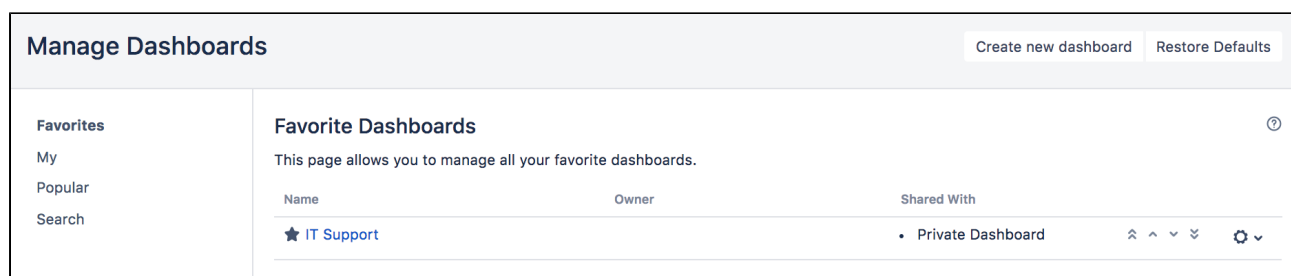
Note on dashboard permissions

Jira administrators, as set inglobal 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.
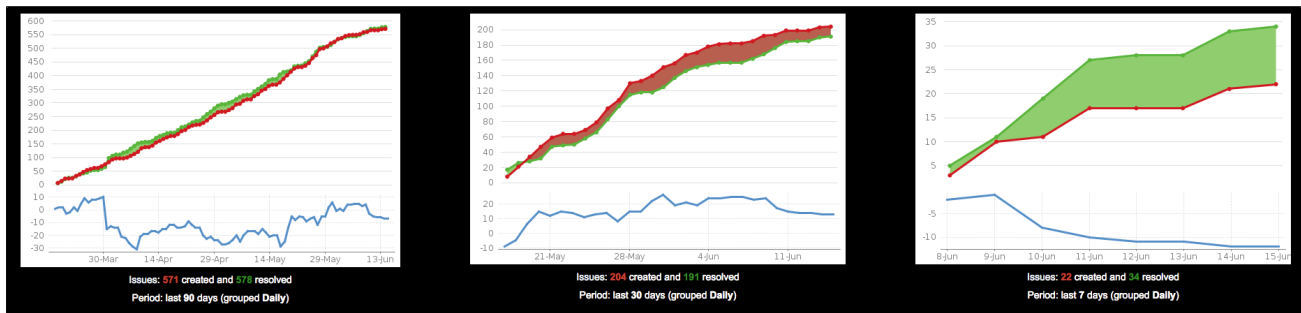
Adding and customizing gadgets

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 Delete a gadget	Use the **More** ( ) button in the gadget header.

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, youll 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	✅	✅	✅	Tracks the correlation of issues in a project or filter during a configured period based on: • the number of days issues have been open • the number of comments in issues • the number of participants or votes in issues The horizontal axis represents the number of days when issues remained open. The vertical axis represents the number of comments in issues. 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. Only the first 200 open issues are displayed on the bubble chart. You can configure the following settings for the chart: • 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	✅	✅	✅	Shows the work progress by showing the number of created and resolved issues for a set period. The chart is based on your choice of project or issue filter. The chart can be either cumulative or show the issue count. An issue is marked as resolved in the time period if its resolution date fits into this period. The resolution date is the last date when the **Resolution** field was set to any non-empty value.
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		✅		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. 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). The gadget doesnt show the progress of work logged in the **Remaining Estimate** and **Time Spent** fields in Jira. 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. If you add or remove issues without estimates, the scope change wont be altered. If you're using Jiras time tracking, the scope change wont show. The **Blocker** field counts all blockers with the statuses **To Do** or **In Progress**. For more information, see Configuring estimation and tracking
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	✅	✅	✅	Shows HTML text on the dashboard.

				⚠️ The Text gadget is planned for deprecation in upcoming releases. To find out about the deprecation, check Jira Software release notes.
				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.
				To enable the gadget:
				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**.
				If you can't enable the gadget, contact Atlassian Support for assistance.

Bamboo gadgets

Bamboo Charts	✅	✅	✅	Shows Bamboo plan statistics in your dashboard. 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.
				When you add the gadget to the Jira dashboard, you may see the following message:
				The website (container) you have placed this gadget on is unauthorized. Please contact your system administrator to have it approved.
				To fix this, you should configure the Bamboo site to allow Jira to draw information from it via gadgets on the dashboard.
				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.

Bamboo Plan Summary Chart	✅	✅	✅	Shows a graphical summary of a Bamboo build plan. If you want to add the gadget to the dashboard, a Jira admin should con figure 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. When you add the gadget to the Jira dashboard, you may see the following message: The website (container) you have placed this gadget on is unauthorized. Please contact your system administrator to have it approved. To fix this, you should configure the Bamboo site to allow Jira to draw information from it via gadgets on the dashboard. 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.
Bamboo Plans	✅	✅	✅	Shows a list of all plans on a particular Bamboo server and the current status of each plan. If you want to add the gadget to the dashboard, a Jira admin should con figure 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. When you add the gadget to the Jira dashboard, you may see the following message: The website (container) you have placed this gadget on is unauthorized. Please contact your system administrator to have it approved. To fix this, you should configure the Bamboo site to allow Jira to draw information from it via gadgets on the dashboard. 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.
Clover Coverage	✅	✅	✅	Shows the clover coverage of plans from a particular Bamboo server. If you want to add the gadget to the dashboard, a Jira admin should con figure 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. When you add the gadget to the Jira dashboard, you may see the following message: The website (container) you have placed this gadget on is unauthorized. Please contact your system administrator to have it approved. To fix this, you should configure the Bamboo site to allow Jira to draw information from it via gadgets on the dashboard. 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.
FishEye gadgets				

|				
Crucible Charts	✅	✅	✅	Shows statistical summaries of your code reviews. If you want to add the Crucible Charts gadget to your dashboard, a Jira admin should configure the FishEye app on your Jira server.
FishEye Charts	✅	✅	✅	Shows Lines of Code data from a FishEye repository. If you want to add the FishEye Charts gadget to the dashboard, make sure that: 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: `subversion`, `perforce`, or `cvs`. Contact the Jira admin about these steps.
FishEye Recent Changesets	✅	✅	✅	Shows two charts about your repository in one: Lines of Code and commit activity. If you want to add the FishEye Recent Changesets gadget to the dashboard, make sure that: 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: `subversion`, `perforce`, or `cvs`. Contact the Jira admin about these steps.

251

Using Jira on a mobile device

> ⓘ This page is about**Jira mobile web**, which is accessed using the browser your your device.
>
> For a richer experience, try theJira Data Center and Server mobile appfor iOS and Android.

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 theJira desktop interface via the mobile menu (shown in the screenshots below).

What does Jira look like on a mobile device?
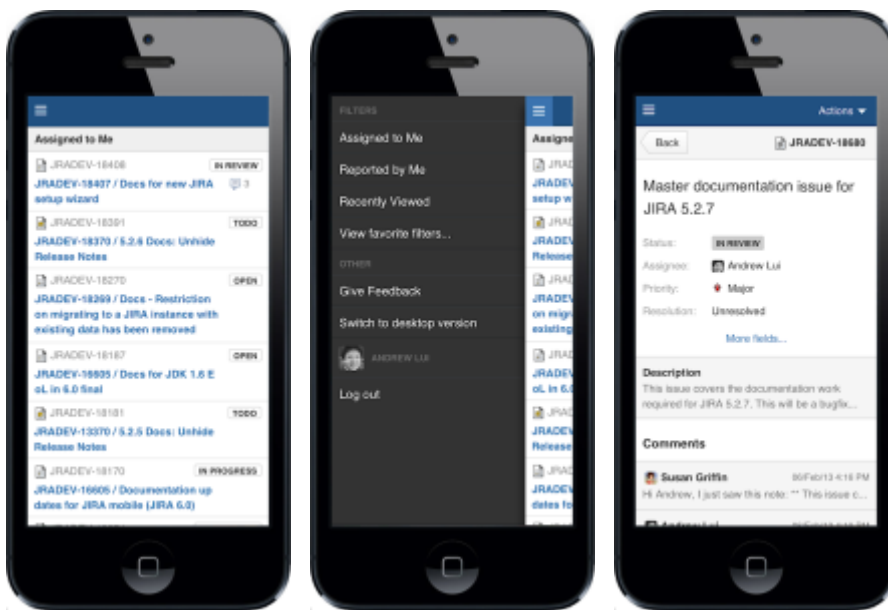
What can you do in Jira on a mobile device?

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).
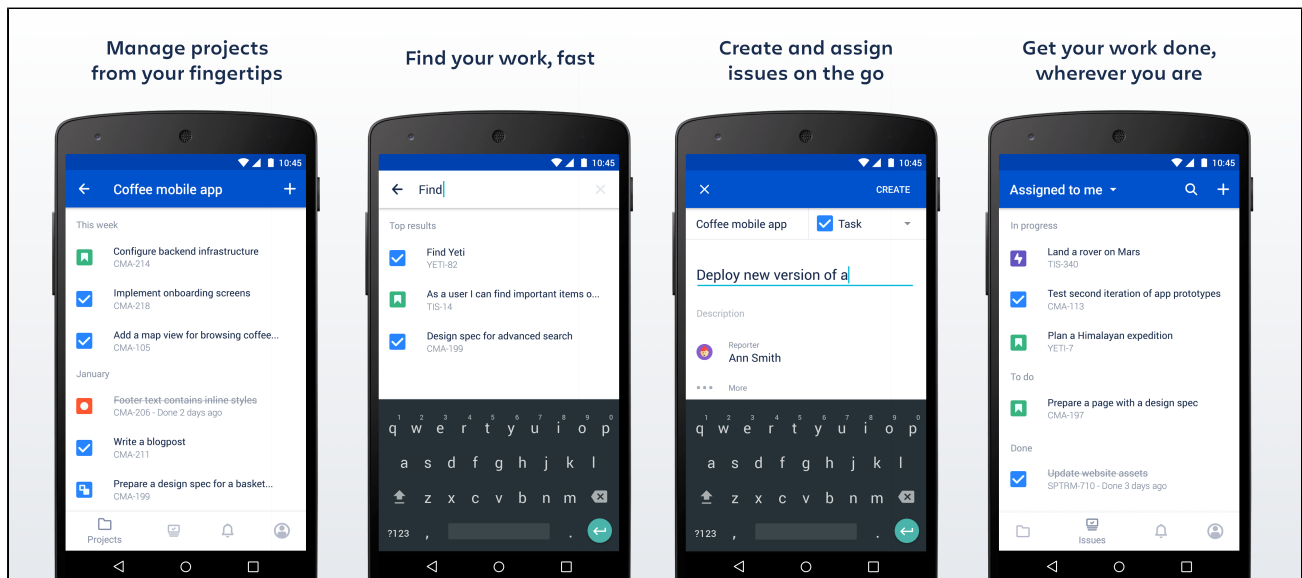
Jira Core mobile app

A native app for Jira Coreis here.

> ℹ️ **Using Jira Software?**
>
> If you're using Jira Software, see Jira Data Center and Server mobile app.

About the app

Here's what you'll get, and what you can do in the app:

- **Manage projects from your fingertips**
  Search and browse your projects, along with all issues your team is working on. View details about your issues and add some more.
- **Collaborate on the go**
  Create issues and add them straight to your projects. Add comments to keep your team in the loop, and@mention away to get your teammate's immediate attention.
- **Get back to your work, fast**
  Use your favorite filters to quickly find important issues and projects, or just view the list of all issues you've been working on recently.
- **Stay up-to-date with push notifications**
  Get real-time notifications about activity in your projects so you don't miss a single thing. Customize what you're notified about and when. Learn more
- **Open links directly in the app**(forJira 8.4 and later)
  To help you navigate between pages, you can go directly from a link, for example in an email notification, to the app. If you do not have the app yet, clicking the link will take you to the app download pagethe App or Play store. Simply click the**Open in app**button that displays in mobile web.

To use this option, make sure to upgrade yourJira to version 8.4 or later.

What you'll need

Jira Core requirements

In order for your users to try the Jira Server app, you will need to upgrade your Jira instance toJira Core 8.3or later.

Device requirements

In order to use the app, your users will need a device with either:

- Android 5.1 (Lollipop) or later, or
- iOS 12 or later (iPhone, iPad or iPod Touch)

Upgrading the app

If you have used the app before and would like to try out the latest version, make sure you upgrade yourJira instance toJira Core 8.3or later.

Downloading the app

You can download the app forAndroid(both the public version and beta are available)oriOS.

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 ourKnowledge base articlefor 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.

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 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 on push notifications, see Push notifications.

Related pages and known issues

Having problems when using the app? Check if they're listed here:

- 'Response status code was unacceptable: 405' error in the mobile app
- 'Can't check compatibility' error in the mobile app
- 'Can't connect to your site' error in the mobile app
- 'We're missing something' error in the mobile app
- 'Can't get a secure connection' error in the app
- How to enable or disable access to the app
- How to find your site URL to set up the app

- Problems with logging in because of missing headers or cookies
- After logging in, the app displays a desktop version of Jira
- Problems with logging in with SAML

For all known issues, seeJira Server and Data Center mobile apps.

Invite your team to use the app

If you're running Jira 8.3or later, invite your team to start using the Jira Server mobile app.

Head toJira Core mobile appto read up on some considerations for administrators, and find out whether you can use the app with your instance.

Users don't need any additional permissions to use the app, you just need to let them know where to download it and how to log in. If your Jira instanceis not accessible on the public internet, you may also need to help them connect to your network or VPN on their device.

Email template

Here's a suggested email template that you can adapt to let your users know that the Jira Server mobile app is available. Don't forget to test that you can connect to your site before sending!

> ⓘ Hi everyone,
>
> Jira mobile app is now available for Android and iOS, and you can use them with our Jira instance. With the mobile app, you can create and edit issues, move them on your board, and track the activity in your projects, straight from your device.
>
> To use the app, you'll need a device with either Android5.0 (Lollipop) or later, or iOS 12 or later (iPhone, iPad or iPod Touch).
>
> Download the app for:
>
> - Android:https://play.google.com/store/apps/details?id=com.atlassian.jira.server
> - iOS (Apple): https://itunes.apple.com/us/app/id1405353949?mt=8
>
> To log in you'll need our Jira URL:**<add your full URL, e.g.** https://yoursite.customer.com/jira **>**
>
> **<If your site isn't accessible outside your network>**
> To use the app you'll need to be connected to our network / VPN.**<Add steps for connecting to VPN, if applicable>**
>
> **<If your certificate is self-signed or from an unknown Certificate Authority>**
> You'll also need to install our certificate on your device.**<Add steps for downloading the certificate>**
>
>
> Best,
>
> Your name here

Mobile Device Management (MDM)

You can distribute the Jira Server app to people in your organisation using your MDM solution. This allows you to:

- Deploy the Jira Server app to company-approved iOS and Android devices.
- Pre-populate your Jira site URLs (just the URL, we don't pass login credentials).People will still be able to enter a URL, which is useful if you have some rarely used sites, and don't want to pre-populate them all.Requires Jira Server iOS app version 1.3.2 / Android app version 0.7.5 or later.

MDM Function	Supported for Jira Server app
App deployment	Yes
App configuration	Yes - to pre-populate the site URL only
App tunnel	Yes
Single sign-on (SSO)	No
Access control	No
Security policies	No

Read more aboutAppConfigand what we currently support in ourAppConfig Technical Capabilitieswhite paper.

Distribute the app to managed devices

The way you do this will depend on your particular MDM provider. In most cases you will:

- Add the app to your MDM app catalog.
    - For iOS you will likely bulk purchase through the Apple VPP store, then link the app in your MDM solution (this might be by providing an sToken, or could happen automatically)
- Choose which devices should be able to install the app (this might be called something like distributing or assigning the app).

Refer to the documentation for your MDM provider for more information.

Pre-populate site URLs on the login screen

If your MDM solution supports the AppConfig standard, you can save your users time and prevent mistakes by pre-populating your site URL in the app.

To pre-populate the mobile app login screen with one site URL:

1. In your MDM, navigate to the App Config section. Check the documentation for your MDM for how to do this.
2. Add a new key called "sites"
3. In the Value field for the key, enter your site title and URL in JSON format as shown in the examples below. Replace the title and base URL with your own site details.

   For a single URL:

   ```
   [
   { "title": "My Jira Site", "baseURL": "https://jira.example.com"}
   ]
   ```
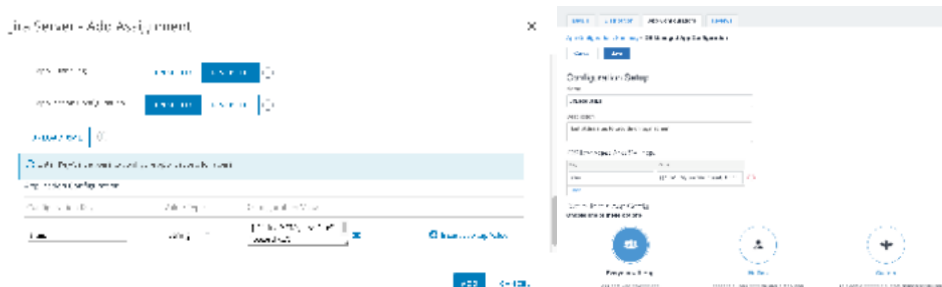
   For multiple URLs:

```
[
 {"title": "My Jira Site", "baseURL": "https://jira.example.com"} ,
 {"title": "Bug Tracker", "baseURL": "https://bugs.example.com/jira"}
]
```

Copy the JSON carefully, including the[and].

4. Save your changes.We recommendyou access the JiraServer app on a device to check you have passed the URLs correctly, before distributing the app config changes to your users. You'll see an error if the app can't display your list of sites.

Here's an example of how the AppConfig looks in AirWatch and MobileIron, two popular MDMs.


Parameters

Parameter	Required	Description
title	Yes	Descriptive name of your Jira instance.
baseURL	Yes	URL of your Jira instance. You can check it in **Administration > System > General configuration**. URL of your Confluence instance. You can check it in **Administration > General configuration**.
skipInfo	No	Use this parameter if youre having problems with logging in because your proxy server blocks any unauthenticated requests. When set to true, the mobile app wont make `login/server-info` calls that require authorization and instead redirect your users to your single sign-on page. In general, the Mobile plugin for Jira is a prerequisite for using the mobile app. However, if youre using the `skipInfo` parameter (set it to `true`) and have the plugin disabled, your users will be able to view a desktop version of Jira and log into it, which is not the correct behavior, but might not look like an obvious error. If youre having any problems with logging in, see our knowledge base articles: Jira Server and Data Center mobile apps (Requires iOS app 1.13.0 / Android app 0.12.1)

Push notifications

Push notifications are a great way to stay in the loop, as they appear on your device, even when youre not using the app. Tap the notification, and be taken straight into the app.
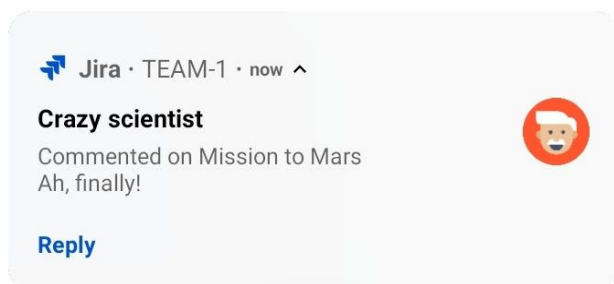
Heres a little push notification as seen on Android.

> ✓ Your Jira admin will need to enable push notifications for your Jira instance in **Administration > System > Jira mobile app.** If they dont, your settings will still apply to in-app notifications.

To change your push notification settings:

* For Android, go to**Account > Settings > Notified about**
* For iOS, go to**Account > Notifications**

You can manage the following notification types:

Type	Notified about	Details
Assigned	Issues you're assigned to	Issue update, including:
Reported	Issues you've created	• Updated description • New attachment
Watching	Issues you're watching	• New assignee • New comment • Issue transition
All mentions	Mentions of you	Mentions of you in: • Issue description • Comment

Good to know

* If your instance isn't accessible on the public internet (for example you need to be connected to your office wifi, or use a VPN to access it from home) we adapt the push notification message, so that you get a shorter version when you're not connected to your network.
* Your admin can disable push notifications for the entire Jira instance. If this is the case, you'll see a message when you go to the notification settings, and the settings you choose will apply to in-app notifications, which you can always view in the**Notifications**tab.

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 thingwe've got you covered.

> ⓘ **More accessibility improvements!**
>
> Accessibility in Jira is not a one-time effortwe'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:
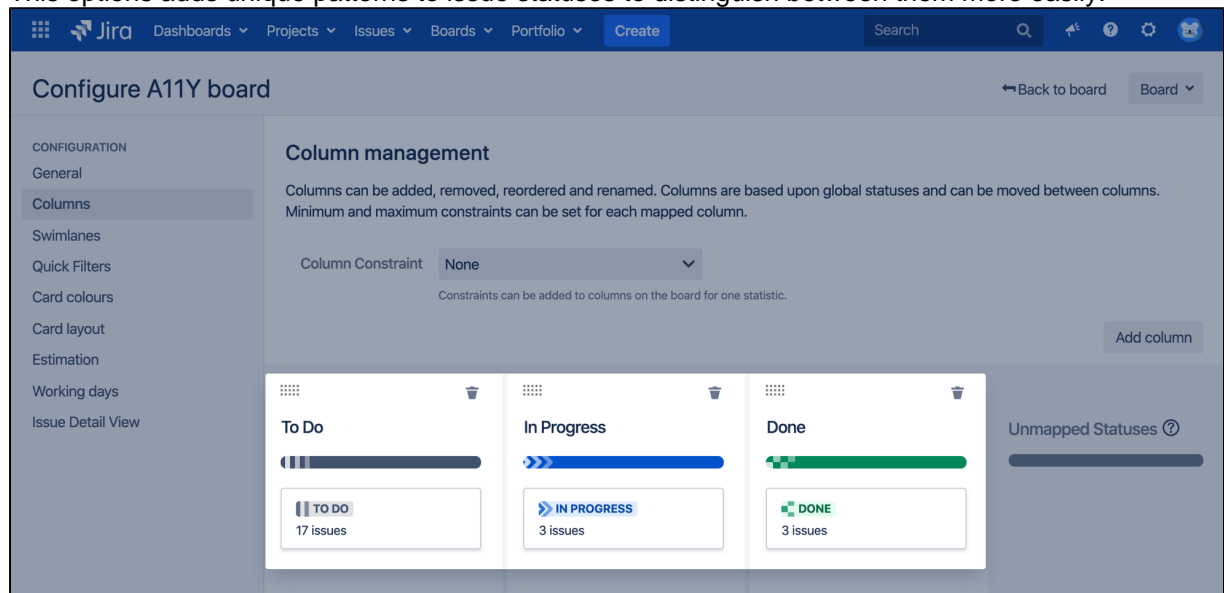
- **Underlined links**
  This option underlines all hyperlinks around Jira to make them more visible.
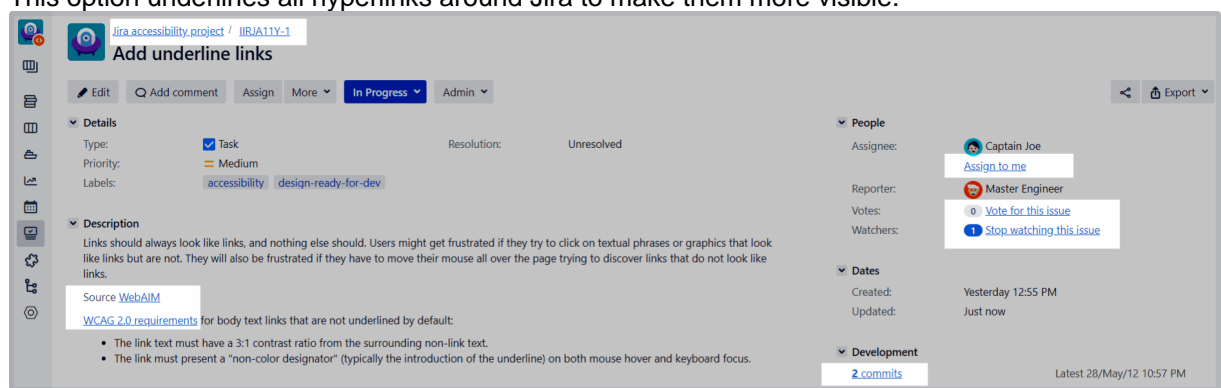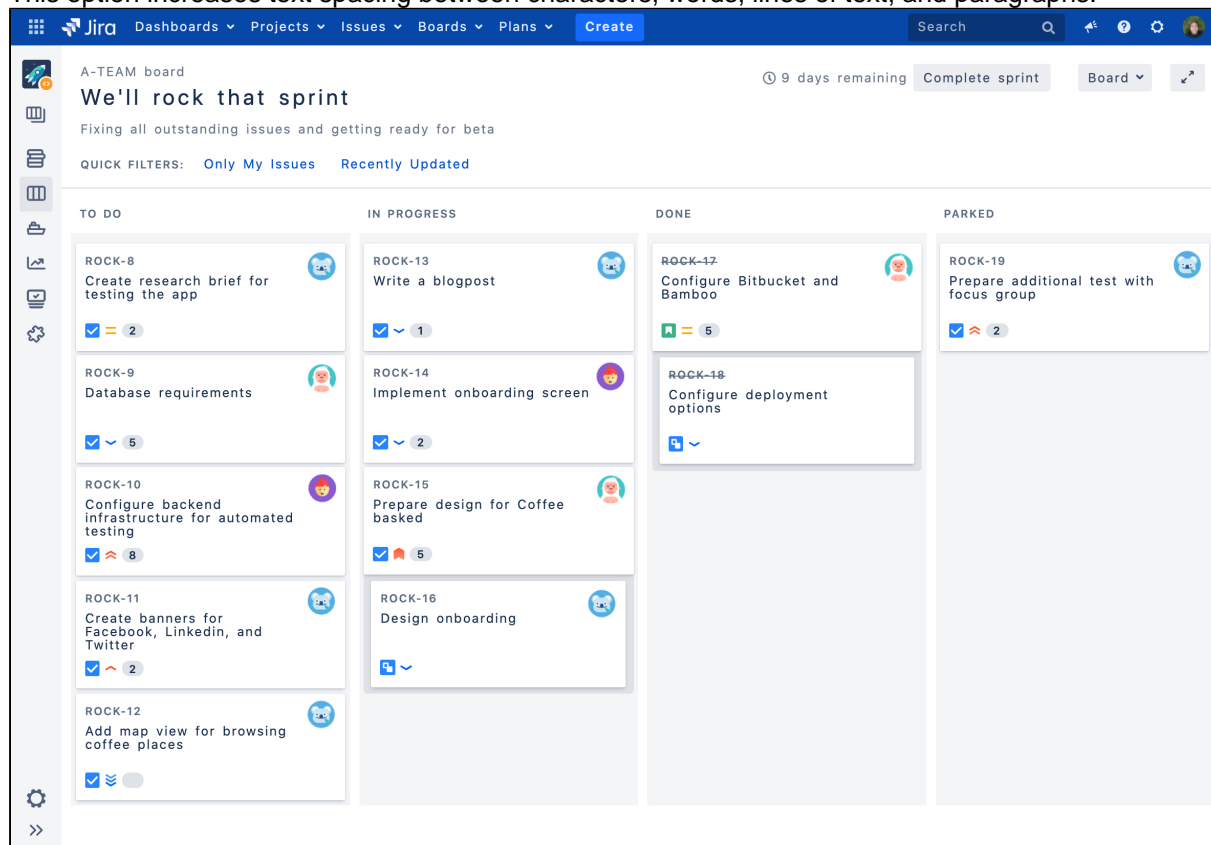

- **Patterns on issue statuses**
  This options adds unique patterns to issue statuses to distinguish between them more easily.
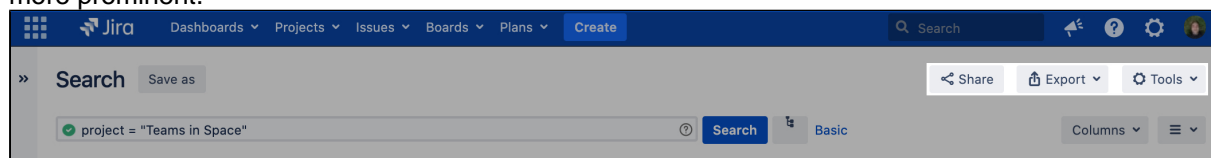
■ **Increased text spacing**
This option increases text spacing between characters, words, lines of text, and paragraphs.

■ **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.

Getting help

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(s).

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 Jiraknowledge 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.

I don't like the way something works

Raise a suggestion*.

Something else?

If you need help with something else, raise a support request*.

** If you are the **Jirasystem 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.

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 Jirainstance or an unsupported JIRA platform,
- you are trying to perform an unsupported operation, or
- you are developing an add-on for JiraCore.

You can also have a look at the most popular JIRA answers and most popular JIRA Core answers.

Jiraknowledge 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 trackerrecords 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.