ATLASSIAN

Jira Software Data Center and Server 9.12

Contents

Jira Software Data Center and Server 9.12 documentation

Jira Software combines development tools with agile features to help teams build world-class software.

Get started

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

View guides

What's new

Time to upgrade? Get the low down on the latest and greatest in Jira Software 9.12.

View latest changes

Jira Software overview

Welcome to Jira Software! This chapter gives you a rundown of the awesome features of Jira Software, as well as how Jira Software fits in with the Jira family of applications.

What is Jira Software?

Jira Software is a powerful platform that combines issue collection and agile project management capabilities into a single application. Using Jira Software helps you plan and organize tasks, workflows, and reports for your agile team more efficiently.

What is Jira Admin?

The powerful issue collection and agile project management capabilities of Jira Software are built on top of Jira Admin. Jira Admin also contains the administrative functionalities of Jira Software. Many application settings are configured in Jira Admin, like creating and editing workflows, or configuring permissions. Whenever you are asked to perform functions like these, which require administrative access to Jira Admin, we will direct you to the Jira administration documentation to accomplish these tasks.

Using this documentation

This documentation will teach you how to set up, configure, and start using Jira Software. We've conveniently divided the documentation into chapters based on typical roles, and the tasks normally associated with those roles.

Leading an agile project	If you are a product manager, product owner, development manager, Scrum master, team lead, or if you hold any role that requires agile project management, then this chapter is for you. The topics in this chapter will help you configure your existing Jira Software instance to suit your agile development process — from starting a new project and building a backlog to releasing a version and preparing reports for your team.
	Learn more...
Working as a team member in an agile project	If you are a developer, designer, QA engineer, technical writer, or anyone else who would use (but wouldn't configure) Jira Software, head on to this chapter. The topics in this chapter will help you use Jira Software to manage your development work — from finding and working on your issues to creating branches for your work and collaborating with your team on issues in Jira Software.
	Learn more...
Administering Jira Software	For system administrators who will set up, configure, and maintain instances of Jira Software . The topics in this chapter will help you administer Jira Software for your agile team — from configuring permissions to managing your Jira Software settings and data.
	Learn more...

Jira Software is a highly customizable application, so your organization's instance of Jira Software may appear different than shown here. In addition, different features and functionality are available to users depending on their permissions. In these instances, we will highlight the permissions required to complete a task.

Jira applications overview

Jira application overview

The Jira family of applications are built on the Jira platform. Jira Core is 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 as Jira Software and Jira 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 as Bitbucket and FishEye, as well as agile boards, but this information is only viewable by a Jira Software user. A Jira Core user would be able to see the Software project, but would not be able to see the Software-specific features, like agile boards or the information from linked development tools. Likewise, a Jira Software user would not be able to see any Jira Service Management application-specific features on a service project, only a basic view of the project and its issues.

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

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

		Jira Core	Jira Software	Jira Service Management

			Jira-Core-user	Jira-Software-user	Jira-ServiceDesk-agent	Customers
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	✗	✗	✓	✗

Getting started with Jira Software

Jira Software unlocks the power of agile by giving your team the tools to easily create & estimate stories, build a sprint backlog, identify team commitments & velocity, visualize team activity, and report on your team's progress.

To give your team a tour through a complete project, the following guide contains two tutorials that show a simple agile workflow at a small software company. This guide touches on some of the most used features and follows the development team at Teams in Space as they work to improve their next generation space travel software.

In this simple workflow, a manager sets up and runs a project while individual users add content and work with issues within a sprint. The following pages are set up to follow this structure and are divided into two tutorials:

- Getting Started as a Jira Software Manager: A guide for managers who set up and run the agile project
- Getting Started as a Jira Software User: A guide for users who work on and resolve issues

Jira Software workflow

The simplified workflow and setup in this tutorial is intended for teams using scrum, so some of the concepts presented may not apply to all teams. Here's how the manager and users work together to complete a sprint:
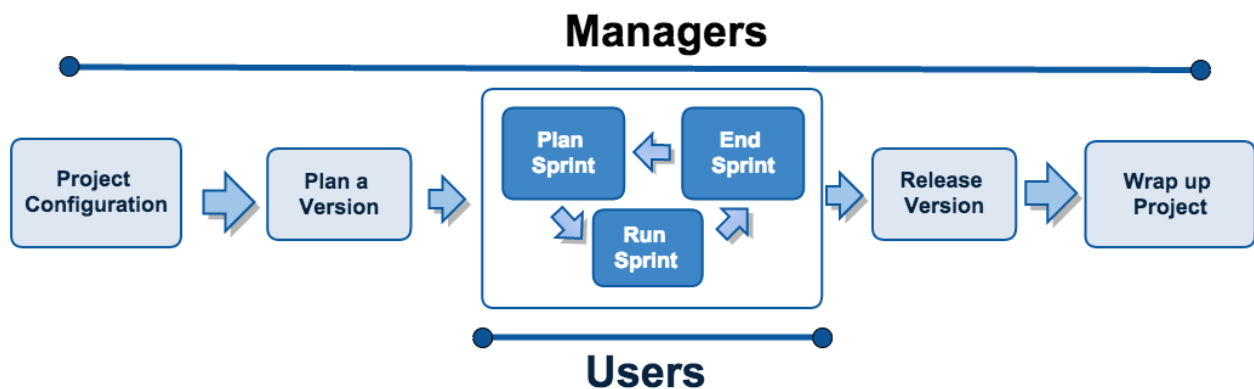
Roles

Typically, variations on the following roles can be found in an agile development environment. We recommend that managers complete both tutorials while users complete the user tutorial.

Jira Software Managers

User with administrative rights for your instance of Jira Software. These roles are usually filled by scrum masters or development managers.

- Access all features in Jira Software
- Create and update projects
- Add and remove users
- Start and end sprints
- Perform other agile tasks

[Get started as a manager](#)

Jira Software Users

User who works on and resolves issues. These roles are normally filled by software developers.

- Create and manage issues
- Check code review status
- Create branches (with integrated development tools)

[Get started as a user](#)

Getting started as a Jira Software manager

Welcome to Jira Software! This tutorial will help you set up your workspace, and create and edit a project for your team. To get your project up and running, we'll also show you how to create, set up, and organize your scrum board.

By the end of this tutorial, you will have:

- Created a new software project
- Added users
- Prepared your backlog
- Started and completed a sprint
- Evaluated the results

But first, let's set up your Jira Software workspace.

Let's go!

Audience: Scrum masters and software development managers that want to streamline their agile workflow!

Time: 1h 30m

Setting up your workspace

1. Setting up your workspace
2. Customizing your project
3. Creating your backlog
4. Grooming your backlog
5. Planning your sprint
6. Tracking your progress
7. Wrapping up your work
8. Doing more with your agile projects

When setting up your Jira Software workspace, you'll need to do the following:

1. Create your project.
2. View your Scrum board.

If you haven't installed Jira Software yet, see these instructions: Installing Jira applications

Create your project

A Jira Software project is a collection of issues and tools that allows your team to coordinate the development of a piece of software. Every project contains configurable boards and workflows that you can create and customize to fit your team's workflows.

1. Log in to your Jira instance. You'll see the System Dashboard, as shown below.
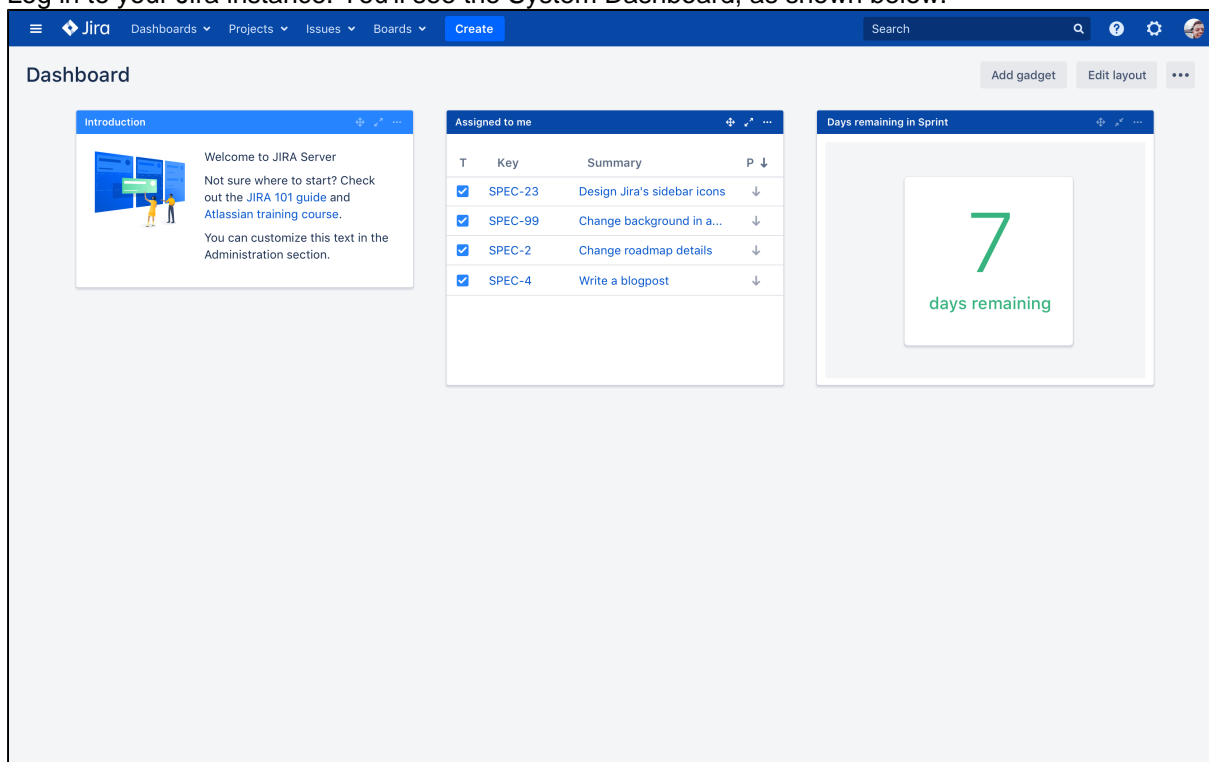
2. Choose **Projects** > **Create project**, and then select your project type. Typically, you would choose 'Scrum software development' for iteration planning, or 'Kanban software development' for constraint-based task management.
   For this tutorial, let's do Scrum since most software developers use scrum in agile projects.
3. Choose **Scrum software development** > **Next**.
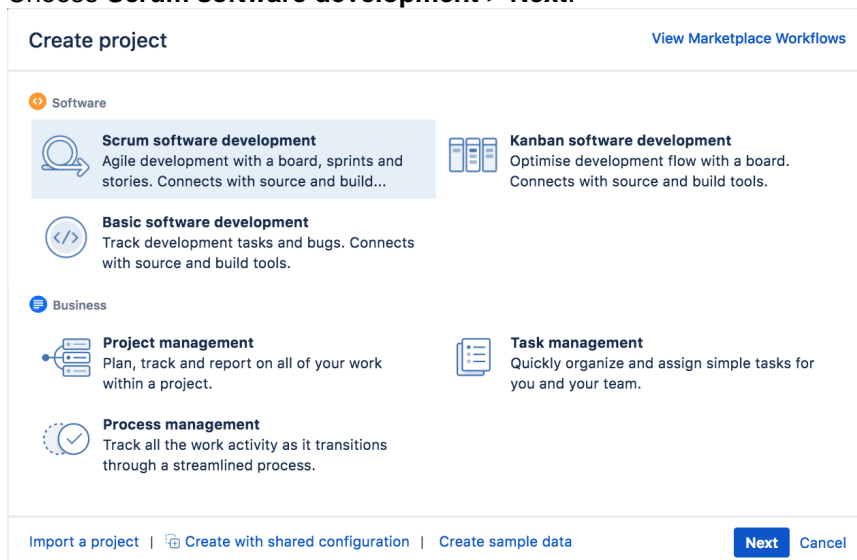
4. In the Name field, enter **Teams in Space**.
5. Choose **Submit** to create your new project.

View your Scrum board

A Scrum board is automatically created with your new project. Use your Scrum board to view and work on issues, such as new features or bugs. You can rank, view, edit, and track issues on your scrum board via the **Backlog**, **Active sprints**, and **Reports**. Don't worry, we'll discuss these three along the way.

This is what your Scrum board will look like:
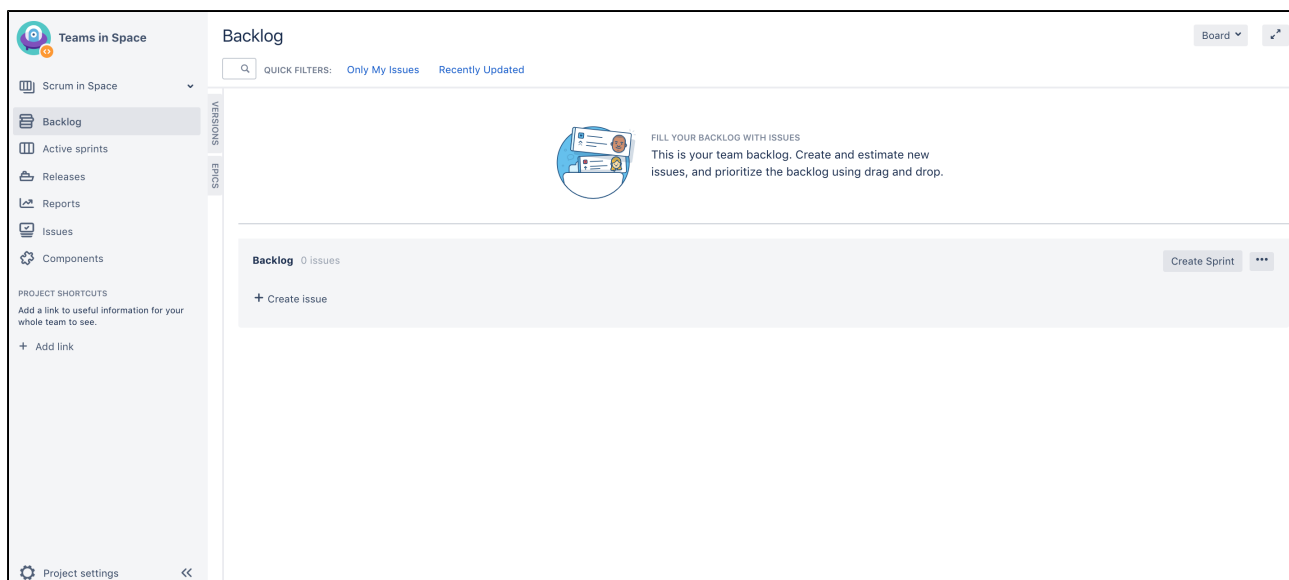
Congratulations, you now know the basics of the Jira Software interface, and have just created your first project! Let's go ahead and customize it!

Next

Customizing your project

Jira Software makes it easy to customize your board to fit your workflow. In this step, let's set up your software team, which will be the Teams in Space team. Teams in Space is an imaginary new company pioneering space travel for innovative travel providers.

Add users

A software team without any members just won't cut it! Let's go ahead and configure Teams in Space's development team. The team consists of you (the manager) and two developers (Jennifer and Kevin).

1. Choose **Administration** (⚙) > **User management**.
2. Select **Users.**
3. In the Create new users page, enter the following details for each user:

Full name	Email address
Jennifer Evans	**jevans@veryrealemail.com**
Kevin Campbell	**kcampbell@veryrealemail.com**

4. Click **Create users**.

Configure estimation and tracking

One of the first things you need to do for a new Scrum project is to configure estimation and tracking. This is essential for understanding how much work your team has and how much it can do, as you build a backlog, run sprints, and review reports.

Scrum teams use different methods to calculate the amount of work involved in completing an issue, and in turn, a sprint. Many teams separate *estimation* (used for measuring the amount of work in the backlog and calculating velocity) from *tracking* (used for measuring burndown of hours used during the sprint), using different units for each. For example, some teams *estimate* tasks in *story points*, then *track* tasks using *hours*.

In this tutorial, Teams in Space uses story points for both estimation and tracking, as per the instructions below.

1. Go to the desired board and select **Board** > **Configure**.
2. Click the **Estimation** tab.
3. In the **Estimation Statistic** field, select **Story Points**. Leave the **Time Tracking** field set to None.

Great! You've successfully added users and configured your project to use story points for estimates. Now, let's work on your backlog!

Previous   Next

Creating your backlog

Your backlog contains tasks for your team to work on in a project. You can add these tasks to a sprint or board so your team can work on them. Since "Teams in Space" is a new project, your backlog will be empty. Let's create some work for your team.

> **How do I get to my board?**
>
> If you navigate away from your board, you can easily return to it by selecting Boards from the top navigation menu, and selecting your board "Teams in Space" from the list of recent boards.

What is a backlog?

Your backlog is usually a list of issues describing what your team is going to do on a project. It's a convenient place for creating, storing, and managing several kinds of issues:

- issues that you're currently working on (you can also see them on the board and in the current sprint if you're using a Scrum project)
- issues which you plan to work on in the near future (for example, in the next sprint)
- issues of low priority which you don't want to forget and want to work on in the future (for example, in a month or two)

The backlogs of Scrum and Kanban projects have different appearances and features. In the following sections, you'll learn more about the differences between them and how to work with both backlogs.

> ⓘ Any project can have only one backlog.

Scrum vs. Kanban backlogs

When you create a Scrum project, a backlog for it is created automatically. In the Scrum backlog, you can organize your sprints, fill them with issues of different types, estimate and prioritize these issues.
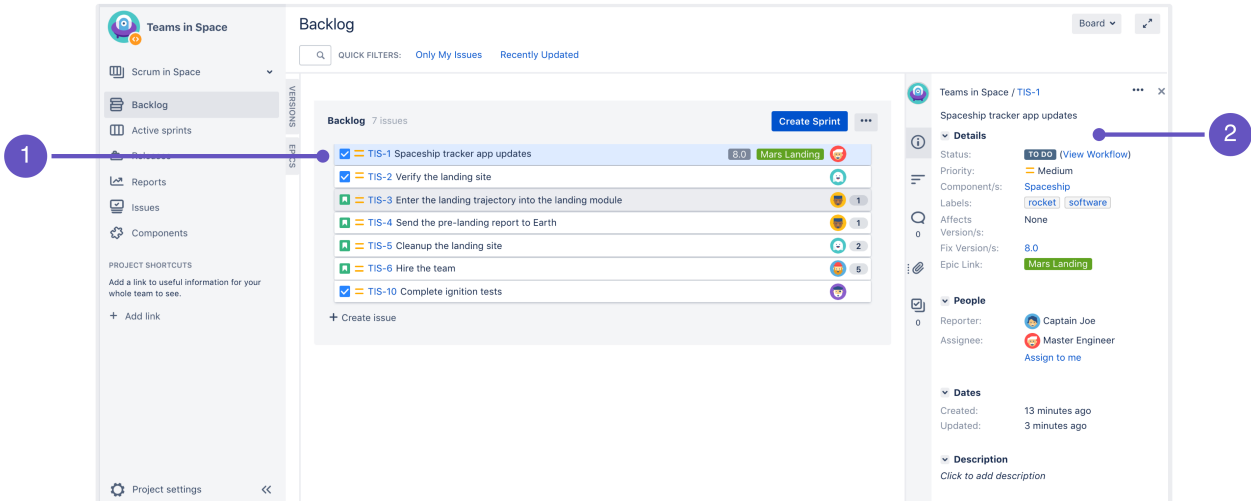
1. **Backlog:** A list of the current issues in the backlog (we've filled it with some issues here to give you an idea of how it'll look like).
2. **Issue detail view:** A detailed view of the currently selected issues.

A Kanban project doesn't have a separate backlog by default. When you create a new Kanban project, you'll see an empty board. The left column on it works as a backlog.

In the following screenshot, see the Kanban board with the Backlog column that contains a newly created issue.
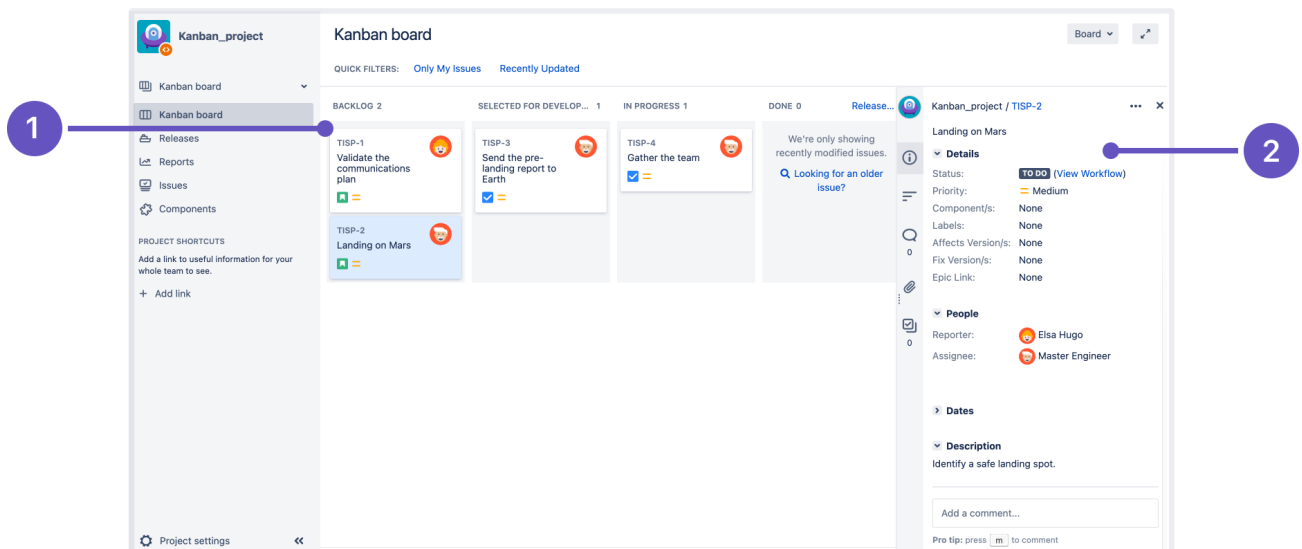
1. **Backlog:** A list of the current issues in the backlog.
2. **Issue detail view:** A detailed view of the currently selected issues.

You can enable a separate Kanban backlog, like in the Scrum project. This is helpful if you have a lot of issues or if you want to keep the backlog separate from the board. See Configuring columns for details.

In the Kanban backlog, you can also add, rank, estimate, and categorize issues. The only functional difference between the Kanban and Scrum backlog is that you won't see sprints in the Kanban backlog, since they aren't part of the Kanban project.

How to find a backlog in an existing project?

You already know that backlogs are created automatically when you create a new Scrum or Kanban project. But what if it's not you who created the project? How to find its backlog then?

In a Scrum project

In the project sidebar, select **Backlog.** If you don't see it, expand the sidebar by clicking the **>>** icon in the lower left corner of the screen.
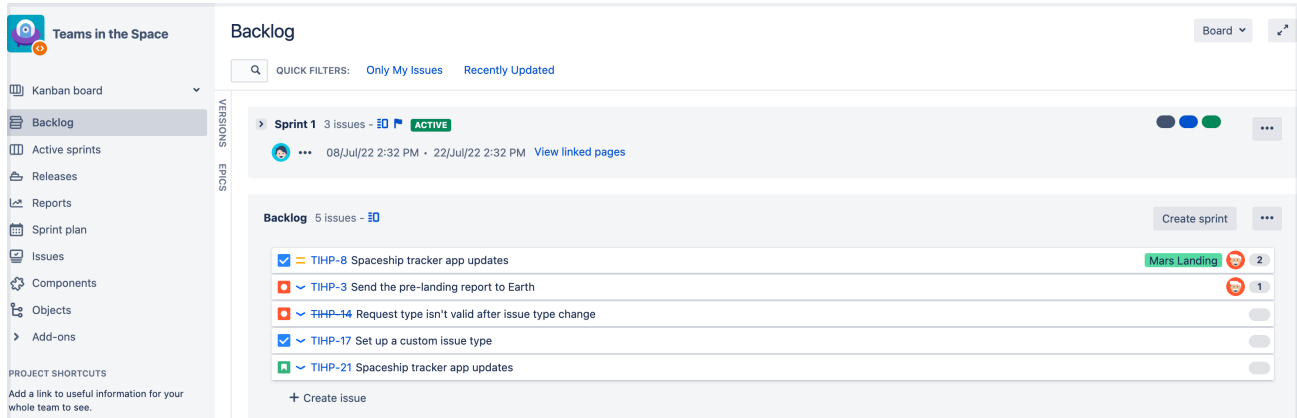
In a Kanban project

If the backlog is enabled, you'll find it in the same way as in a Scrum project.

There may be a case where you'll see neither the **Backlog** column on the board nor the **Backlog** option in the left sidebar. This may happen when the Backlog column is removed from the board, but the separate backlog isn't enabled.

In this case, to find the backlog:

1. In the peoject sidebar, select **Issues**. You'll see the list of all open issues in the project.
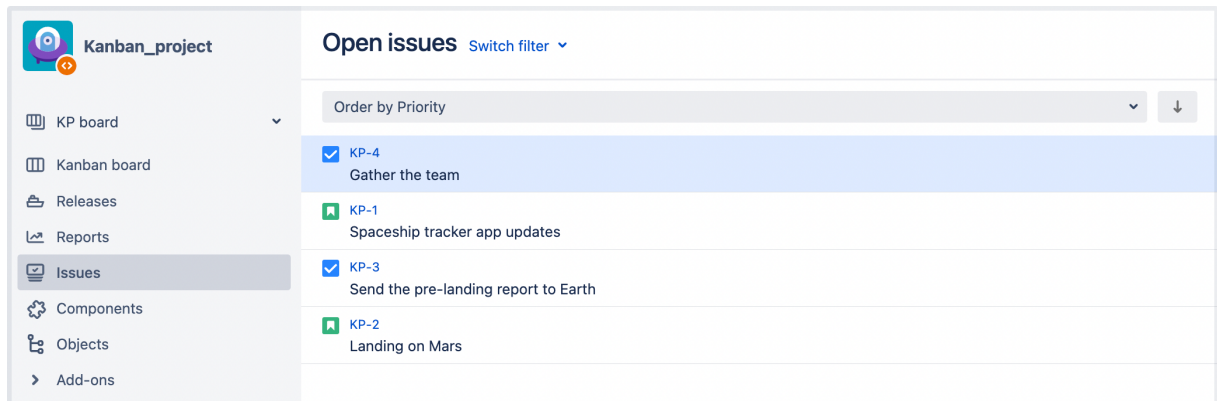
2. In the upper right corner of the screen, select **View all issues and filters**.
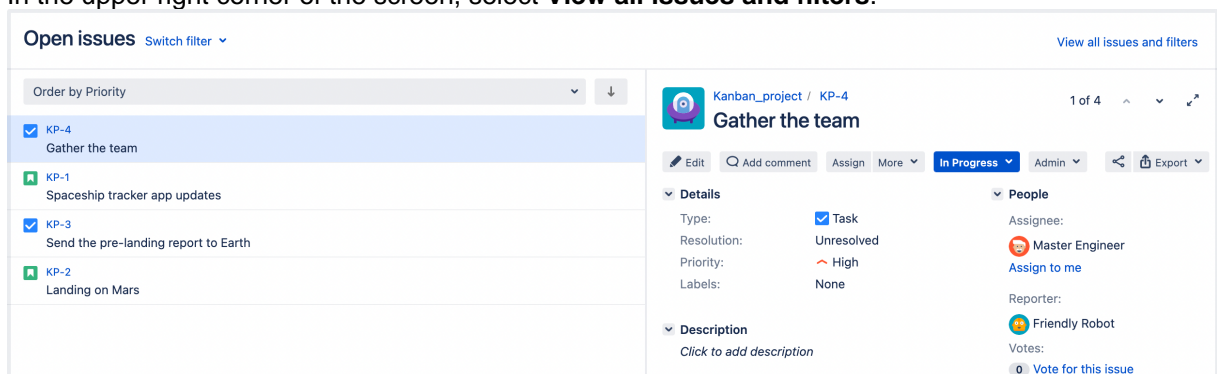
3. In the **Status** dropdown, select **Backlog**. Jira will filter out the issues in your backlog.
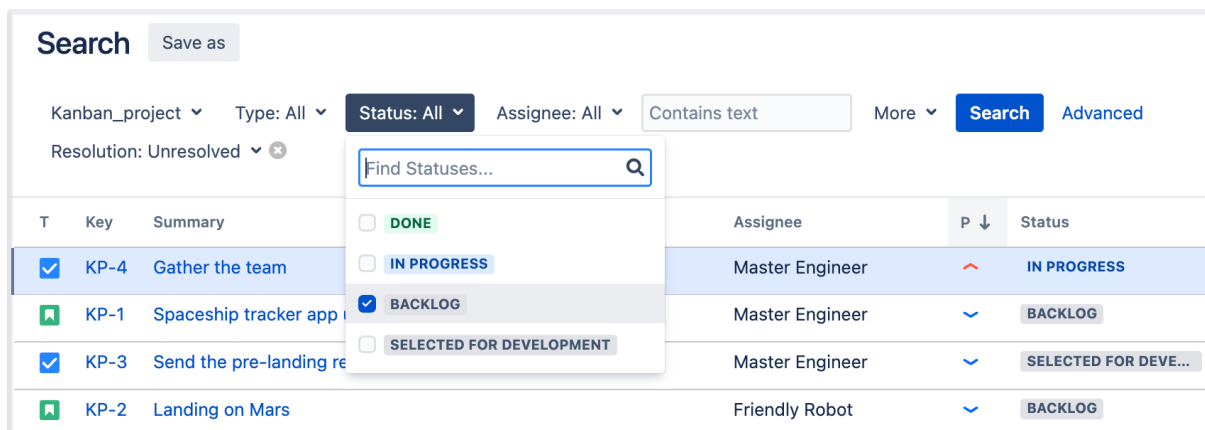
Using a backlog

For a Scrum project, a backlog is created automatically. Learn more about it in Using your Scrum backlog.

For a Kanban project, you have the Backlog column on the board by default. All new issues you create will fall into this column. But you can change this by enabling the backlog in the Kanban project. Learn the details of how to do it in: Using your Kanban backlog.

Adding issues to your backlog

One issue isn't enough to get your team working! Let's add some test issues so you can create and run a complete sprint.

Type	Summary	Priority
Story	Build out a local office on Mars	High
Story	Add support for teams larger than 20 people	High
Story	Next Generation version of SeeSpaceEZ travel platform	Low
Story	Plans for our Summer Saturn Sale	High
Story	Make working with our space travel partners easier	Low
Story	500 Error when requesting a reservation	Medium
Story	Requesting available flights is now taking > 5 seconds	Highest

To create an issue

By default, any team member can create issues. In this tutorial, however, you will create all of the backlog issues.

1. On the Teams in Space board, choose **Create** at the top of the screen.

Tip! If you select the **Create another** checkbox, Jira Software will create your issue and automatically pre-populate a new 'Create Issue' dialog box with your previous issue details.

19

Fill in the fields using the data shown below. Only the fields with * are mandatory.

- **Project:** Teams in Space
- **Issue Type:** Story
- **Summary:** Expand travel to destinations outside of The Solar System
- **Priority:** High
- Leave all other fields blank or at their default values.
2. Select **Create** to make a new issue. An issue key (**TIS-1)** is created for this issue, which comes in handy when searching for issues later.

Learn more about how to create and work with issues in Creating issues and sub-tasks.

Great! Your backlog's all set! Now, let's plan some work for your developers to do!

Previous   Next

Grooming your backlog

An essential part of agile is regularly "grooming" or reviewing the contents of your backlog, particularly before starting any new work.

Adjust your backlog

Before starting your sprint, you need to prepare your backlog. You can easily adjust your backlog by:

- Right-clicking on issues to view, estimate, or add details
- Ranking your issues by dragging and dropping
- Creating new issues
- Editing an issue using the issue detail view

Estimate issues

Now, let's add some estimates to the issues in our backlog. This way, you can easily determine what you can accomplish, and your team can also have a way of measuring the success of the sprint.

1. On the Teams in Space board, select **Backlog**.
2. Select each issue on the left-hand side of the screen to display the issue details on the right-hand side of the screen.
3. Click the **Estimate** field on the right-hand side of the screen for each of the issues, and enter the following information for each issue:

Issue	Estimate
Expand travel to destinations outside of The Solar System	**6**
Build out a local office on Mars	**2**
Add support for teams larger than 20 people	**4**
Next Generation version of SeeSpaceEZ travel platform	**3**
Plans for our Summer Saturn Sale	**8**

Make working with our space travel partners easier	**16**
500 Error when requesting a reservation	**6**
Requesting available flights is now taking > 5 seconds	**4**

Rank the backlog

By default, the issues in your backlog are ranked in the order in which you added them. You can change the rank of your issues according to their relative priority. This helps you organize the issues in your backlog more effectively.

Rankings let you determine whether an issue is more important or urgent than another issue. For example, you may have two separate issues that are both of 'High' priority. Using Jira Software ranking, you can assign one of the issues a higher ranking than the other.

1. Find issue **TIS-8** in your backlog. This issue has the 'Highest' priority, and therefore should be at the top of your backlog.
2. Select **TIS-8** and drag it to the top.
3. Move issues **TIS-2** and **TIS-5** to positions two and three in the backlog. These issues have 'High' priority, but they're not as high a priority as TIS-8.
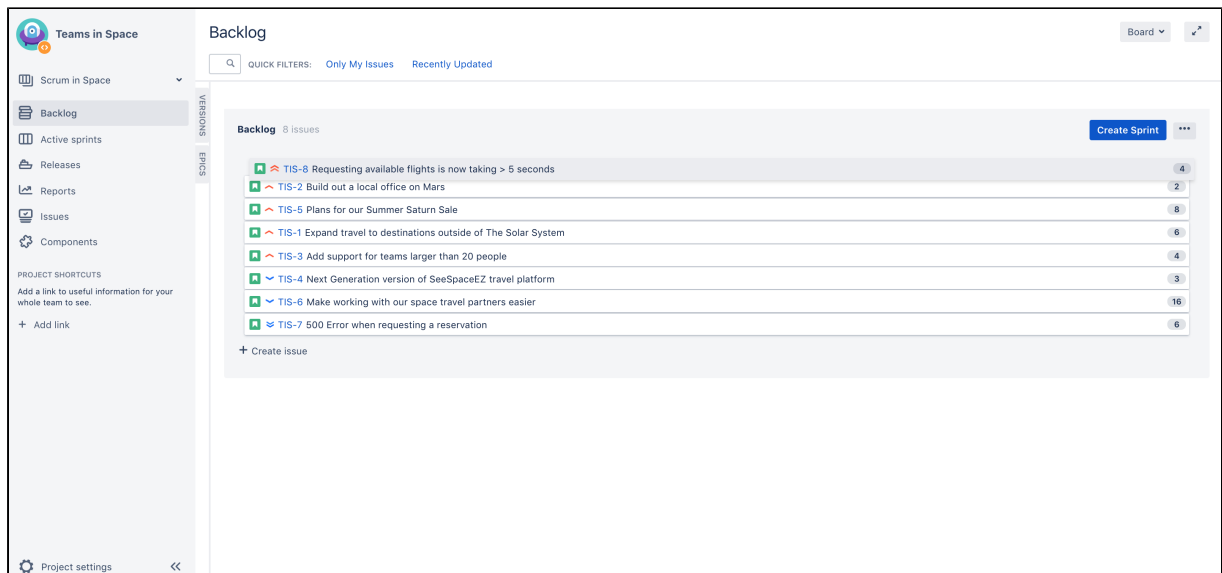
Great! You've just groomed your backlog. Now, let's plan out the details of your sprint!

Previous   Next

Planning your sprint

A sprint is a short period (ideally two to four weeks) during which a development team implements and delivers a discrete product increment, e.g. a working milestone version. In this tutorial, your team will be working in two-week long sprints. Let's go ahead and create a sprint for your team.

Sprint planning

Before creating and starting a sprint, your Scrum team would typically hold a sprint planning meeting. In this meeting, your team should:

- Review the estimates for selected issues
- Break down the selected issues into an initial list of sprint tasks
- Consider upcoming employee time-off, holidays, and other issues that may impact the completion of these sprint tasks
- Gauge the team's capacity team to complete these sprint tasks

By the end of the meeting, your team should be confident enough to commit to completing the work in the sprint.

In this tutorial, we will assume that the Teams in Space team can handle 20 story points of work in a sprint, and that everyone is available for the full sprint. Typically, you would know how much work your team can complete in a sprint by reviewing information from past sprints, usually through velocity and sprint reports.

Need help scheduling?

By using Jira Software in tandem with Confluence, you can embed your sprints using Team Calendars for Confluence. This helps you see the duration of your sprint and how your team's availability or other team events could impact the sprint.
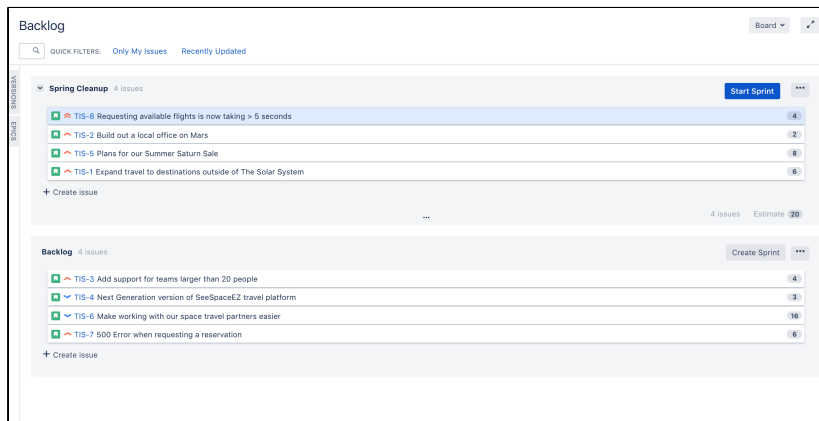
Create a sprint

1. On the Teams in Space board, click **Backlog**.
2. Click **Create Sprint** at the top of the Backlog.
3. Your new upcoming sprint will be added to your board, below any other future sprints. Select the **Sprint 1** text and edit the name of the sprint to 'Spring Cleanup'.
4. The top 4 issues in the Backlog are equal to 20 story points. This is what the team estimated that they could accomplish in the upcoming sprint. Drag and drop the top four issues from the Backlog into your

new sprint.

Start your sprint

Now that you have created a sprint, you can go ahead and start it.

1. Click **Start Sprint**.
2. Today's date and current time become the start date and time for the sprint. For the purpose of this tutorial, enter an end date of 5 minutes from the start date and time.
3. Select **Start** to start the sprint and move the issues into Active sprints.

Congratulations! You've successfully got Teams in Space up and running! Now, let's look at Active sprints to track your team's progress.

Tracking your progress

During your sprint, you and your team will need to monitor your progress to make sure that everyone is on the same page. There are several tools that you can use to do this, which are described below.

Active sprints

The **Active sprints** page is where you monitor the progress of your team's work during a sprint. Here, your team can transition issues through a series of columns (statuses), allowing everyone to quickly visualize the progress being made in the sprint. You can also edit issues by adding information, such as descriptions, attachments, comments. and information from your linked developer tools such as Bitbucket and Bamboo.

More about Active sprints

Issues that you ranked higher in your backlog appear at the top of the column, to make it easier for users to select the most important issue to work on.
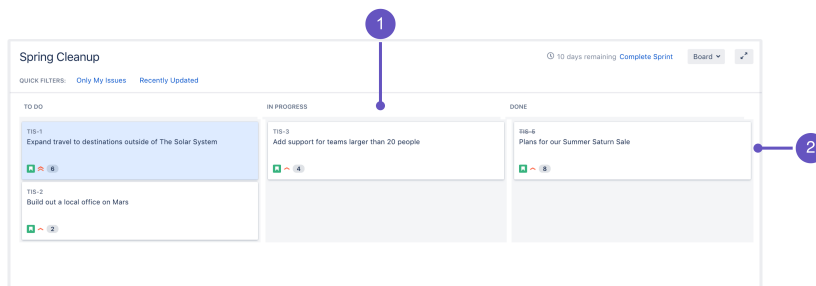
1. **Columns:** A column displays the current status of your issues in your workflow.
2. **Swimlanes:** You can configure swimlanes (rows) using filters, such as stories, epics, assignees, etc.

Transition issues

During the sprint, let's say your team was only able to finish three issues, TIS-1, TIS-2, and TIS-5. Issue TIS-8 remains in the 'To Do' column since no team member was able to work on the issue. Let's show on the board what happened during the sprint by transitioning issues from one column to another.

1. On the Teams in Space board, click **Active sprints**.
2. Select **TIS-5** and move the issue to the 'In Progress' column.
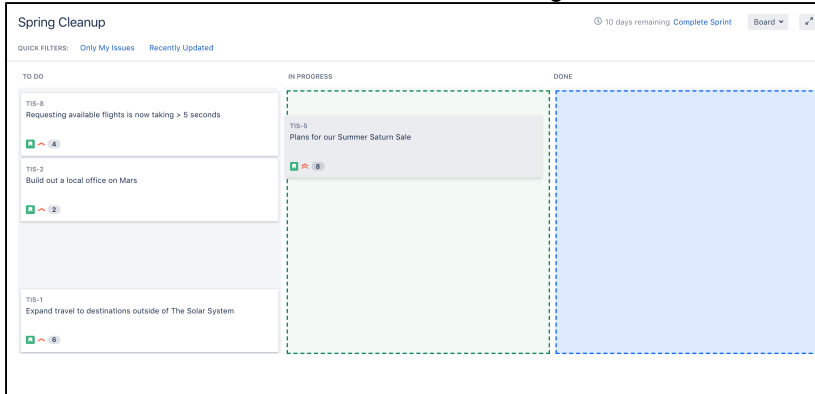
3. Select **TIS -5**, **TIS-2**, and **TIS-1**, and then move the issues to the 'Done' column.

View the Burndown Chart

You've just seen how your team is progressing, from the Active sprints of your board. The Burndown Chart is another useful tracking tool, which can help you visualize your team's progress, as well as determine whether your team is on target to achieve the sprint goal.

1. On the Teams in Space board, click **Reports**.

2. Select **Burndown Chart**.

The grey line in your Burndown Chart is a guide showing the rate of work required to complete the sprint. The red line, on the other hand, shows the actual work completed by your team.

If your Burndown Chart shows the red line above the grey line, your team may not achieve the sprint goal. You may want to consider removing some issues from the sprint. Any changes to scope (e.g. issues added to sprint, issues removed from sprint) are shown in the table below the graph.

View the Burnup Chart

The Burnup Chart provides a visual representation of a sprint's completed work compared with its total scope. It offers insights on your project's progress, as well as offers warnings to help you maintain your project's health; you can instantly identify problems such as scope creep or a deviation from the planned project path.

Burn-up charts has one important advantage over the Burndown Chart in that it allows to divide the scope and the progress. Both scope and progress are combined on the Burndown chart and it is not possible to visualize and identify changes in scope and/or progress.

The chart shows the red work scope line and the green completed work line (completed stories, tasks, fixed incidents for example) in relation to the grey Guideline, which is a theoretical line showing the daily completion necessary to meet the deadline.

The distance between the two lines is the amount of remaining work. The project will be completed when the green and red lines meet.

The spikes in work scope mean that something has been added to the milestone while spikes in completed work mean a story has been completed. This information can be crucial at retrospective to understand if, for example, too much work has ben added or the work has been more complex than estimated.

Well done! You now know how to track the progress of your team's sprints! Next, let's wrap up your work!

Previous   Next

Wrapping up your work

After your team finishes the work in a sprint, you and your team can then perform a retrospective of the sprint. Sprint retrospectives help determine where the team succeeded, and where improvements can be made. The more sprints completed by your team, the more data you can use to find significant areas for improvement.

End the sprint

Once your team reaches the end date of the sprint, you need to end the sprint — regardless if this means some issues in the sprint are not yet completed.

1. On the Teams in Space board, click **Active sprints**.
2. Select **Spring Cleanup** from the Active sprints drop-down.
3. Click **Complete Sprint**. The Complete Sprint dialog box will be displayed, showing the number of issues that are completed in the sprint, and the number of issues that were not completed.

You can move the incomplete issues to the backlog, or to a new sprint. If you had more sprints planned, you could also move them to the next planned sprint. Note, you can always add issues that are returned to the backlog to another sprint later, if you wish.

View the Sprint Report

After a sprint, your team can hold a sprint retrospective meeting, to determine your wins for the sprint, as well as point out the potential areas of improvement that your team can tackle in future sprints. You can use the Sprint Report during sprint retrospective meetings to do this.

The Sprint Report shows a status list of issues in each sprint. It also provides a breakdown of the progress, status, and estimation information for each issue. You can also use the Sprint Report to perform progress checks in the middle of a sprint.

> **Looking for more reports?**
>
> There are many other reports available for your team to use in the Reports page. You can also create your own reports, or use reports in the Atlassian Marketplace.

1. On the Teams in Space board, click **Reports**.
2. Select **Sprint Report** from the Reports drop-down.
3. Select **Sprint 1** from the Sprint Report drop-down.
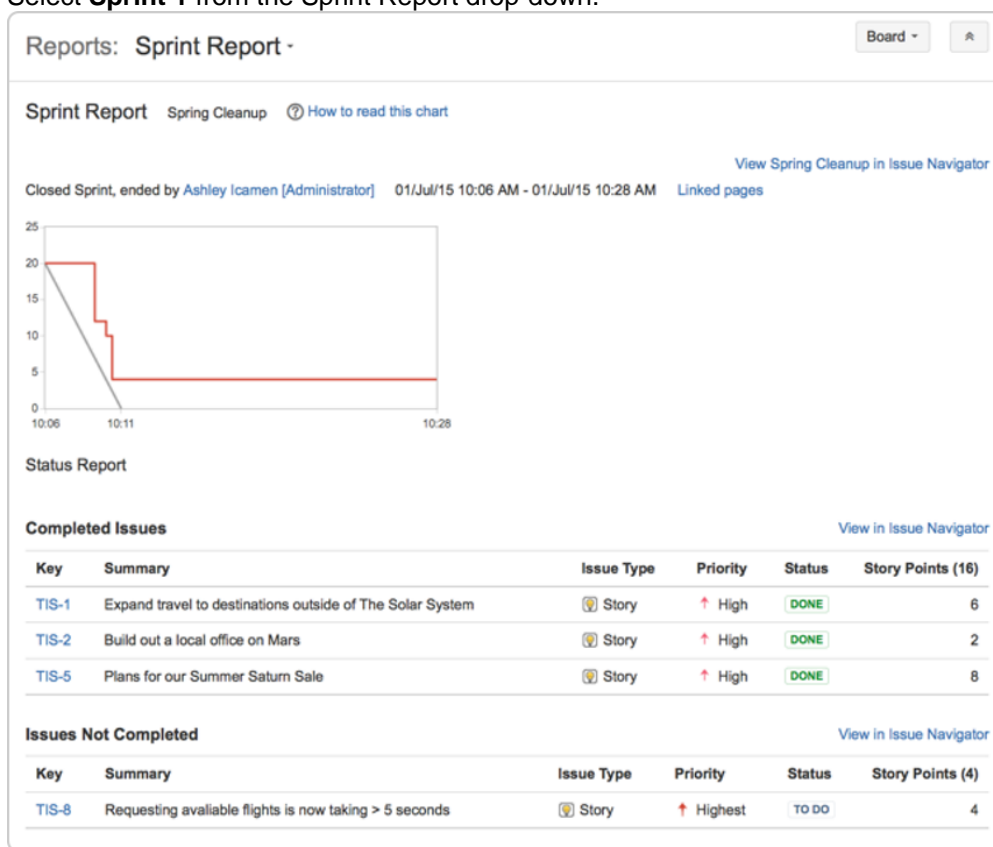
Other tools you can use

Other useful tools can give your team a better visual of the improvement areas – and more importantly, these tools can help your team figure out action plans for these areas. For this tutorial, some of the tools you may want to use are:

- Velocity Chart
- Release Hub

Velocity Chart

You can use the Velocity Chart to track the amount of work your team completes from sprint to sprint. Using the Velocity Chart lets you predict a more realistic amount of work that your team can commit in future sprints.

1. On the Teams in Space board, click **Reports**.

2. Select **Velocity Chart** from the Reports drop-down.
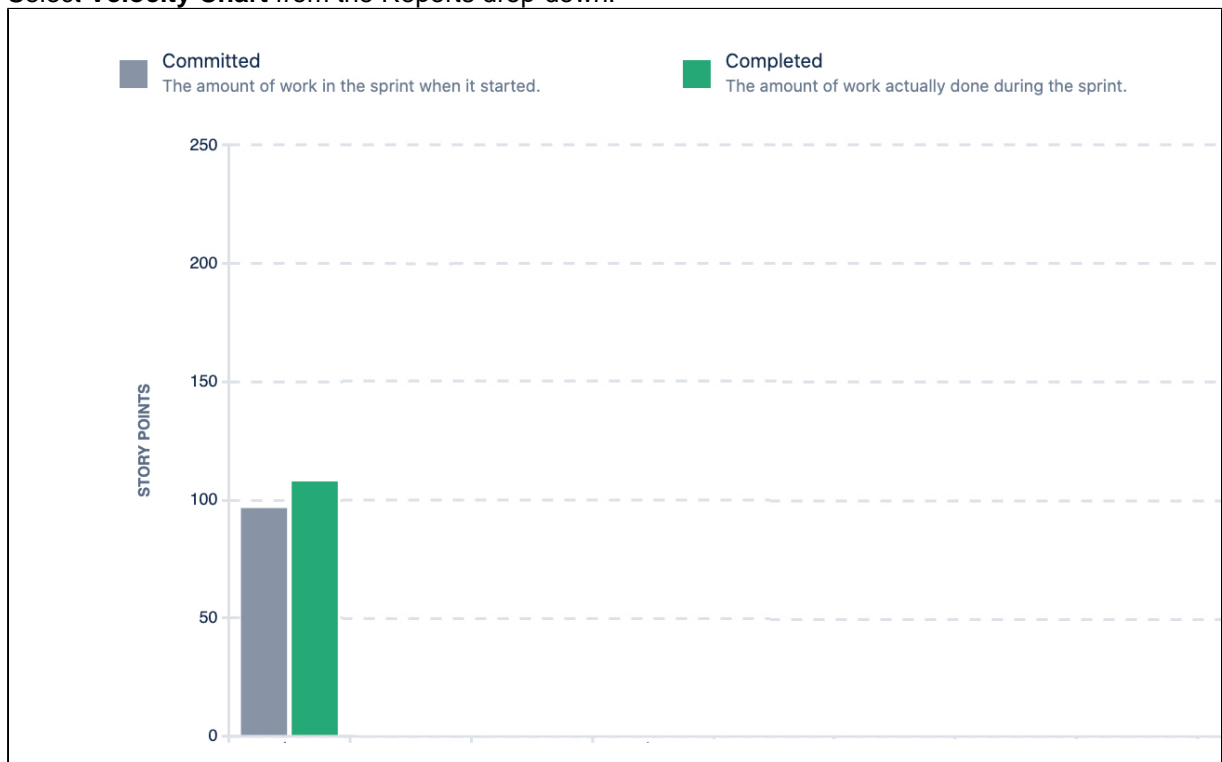
Release Hub

Aside from using reports (like the Sprint Report and Velocity Chart), you can also monitor the progress of a version after you complete a sprint. Monitoring a version's progress helps you see problems early, as well as determine the likelihood of releasing a version on time.

One more step to go! Let's look at some tips and tricks to help you get the most out of Jira Software.

Previous   Next

Doing more with your agile projects

Want to become a Jira Software ninja? Take a look at these advanced topics!

Workflows

A workflow is a set of *statuses* and *transitions* that an issue goes through during its lifecycle. Jira Software makes it easy to create and edit workflows that fit your team's needs. Click here for more information on creating and editing workflows.
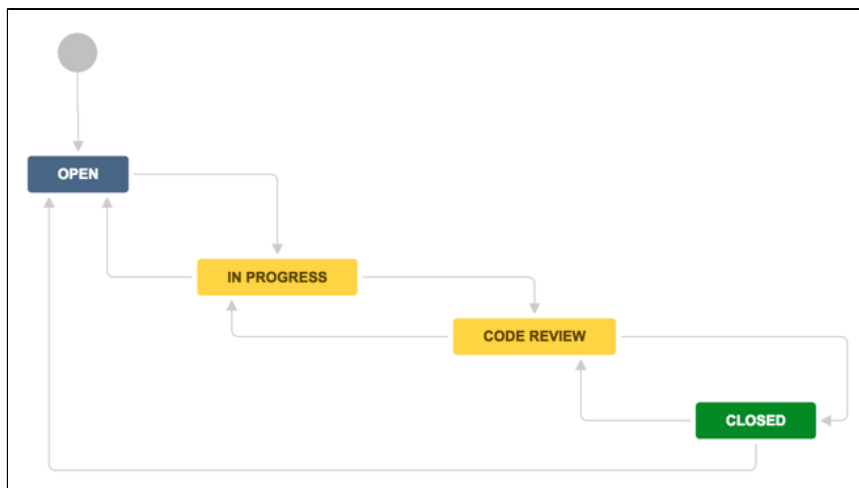
Development panel

Jira Software can be connected to a range of development tools, such as repository managers, code review applications, and build and deployment managers. These tools can help you keep your project tracking in sync with your development work. Click here for more information on integrating development tools.
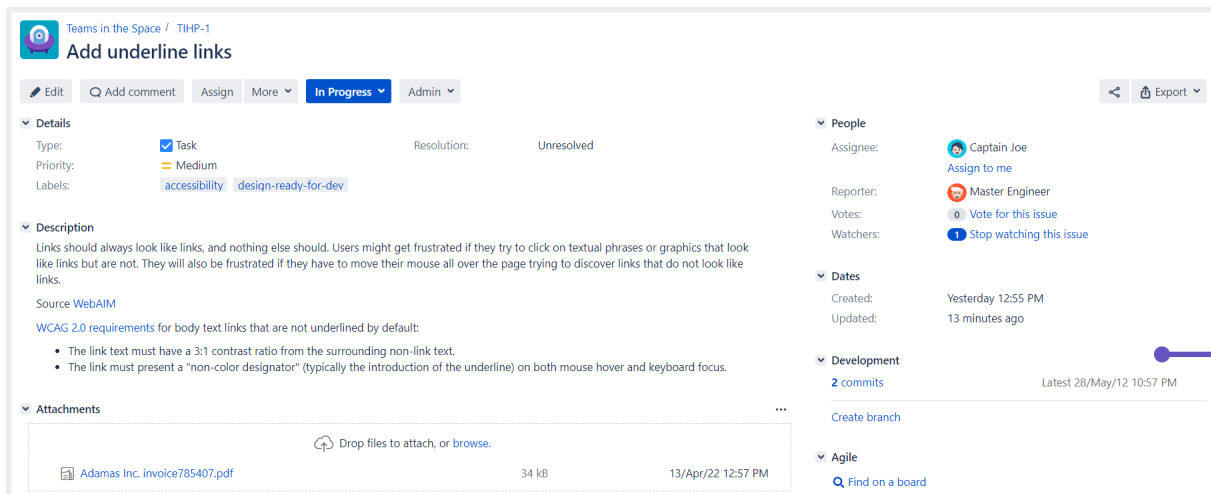
1. **Development panel**

Notifications

Email other Jira users a link to an issue either by sharing the issue with them, or by mentioning them in an issue's **Description** or **Comment** field. See Editing and collaborating on issues for more information on notifications in Jira Software.

Custom cards

Customize issue cards for your boards to bring the right information to your team's attention at a glance. You can change the card colors to help people quickly identify cards on your board as being of a particular issue type, priority, assignee, or any JQL that you want. Click here for more information on custom cards.

Confluence integration

You can link your Jira Software instance to a Confluence instance to create and link Confluence pages to your sprints and epics. For example, you can link your sprint meeting notes in Confluence directly to the relevant sprint. This makes it easy for your team to quickly share information about the sprint. Click on sprints or epics for more information on linking your project to Confluence.
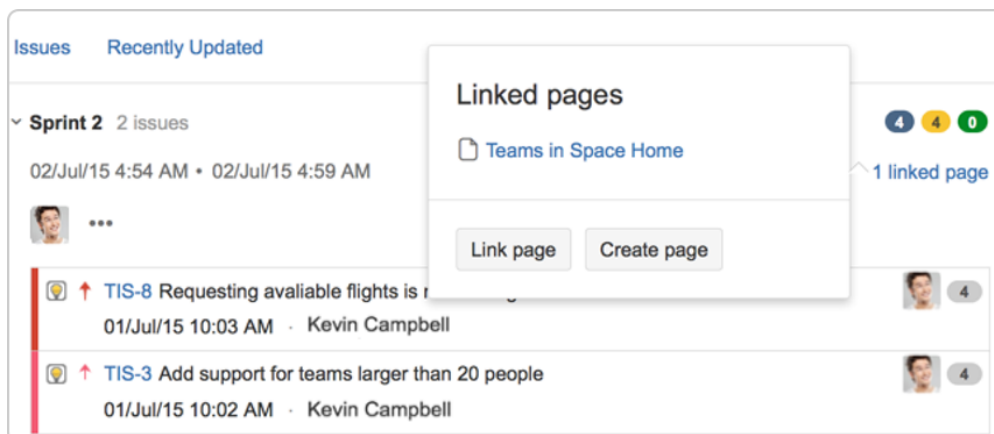
Wallboards

A wallboard runs on a wall display and is used to monitor vital data about a project's progress. You can use wallboards to share information with your team during stand-ups. You can also add gadgets to a wallboard – make sure to add gadgets that are useful to your team, such as a gadget that displays the status of a sprint, including issue status, time remaining, and build status. Click here for more information on creating wallboards and setting up other gadgets for Jira Software.
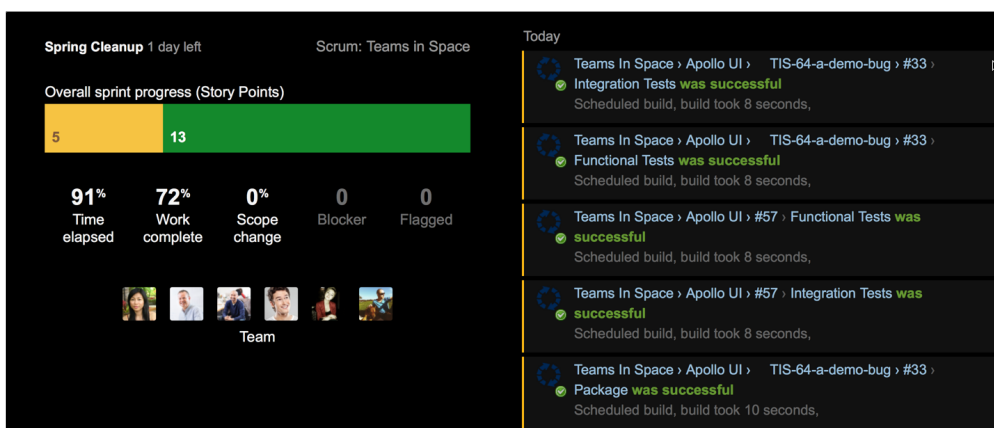
What next?

So that's it — we hope this guide has helped you get a feel for Jira Software. You can continue your training by completing the Getting started as a Jira Software user tutorial, checking out our documentation, or visiting our community on Atlassian Answers for more information.

Previous

Getting started as a Jira Software user

Welcome to your Jira Software project! In this tutorial, we'll introduce you to your project and walk you through a simple workflow – from finding an issue assigned to you, to completing an issue. By the end of this tutorial, you will have completed the following steps as a JIRA Software user going through a simple workflow:

Find your issue | Work on an issue | Collaborate with your team | Create a branch | Start coding | Send your work for review | Finish your work

Find your issue

Typically, your Development Manager will have already created a Scrum or Kanban project, and populated the backlog with issues. There are a number of ways to navigate around your project, but the easiest way to view information is by using boards. A board displays issues from one or more projects – it gives you a flexible way of viewing, managing, and reviewing your work. In this step, we describe how you would typically find an issue to work on, in the active sprint of the board of a Scrum project.

1. Select **Boards** from the top menu, and then select a board from the drop-down list.
2. Select **Active sprints** to see the issues in the sprint of the selected board.
3. Issues are represented by issue cards in the Active sprints of your board. The cards show quick reference information, like issue keys, assignees, and descriptions. Locate the top card in the **To Do** column and drag it to the **In Progress** column. By doing this, you are actually moving to assign the issue to yourself, and start working on it.
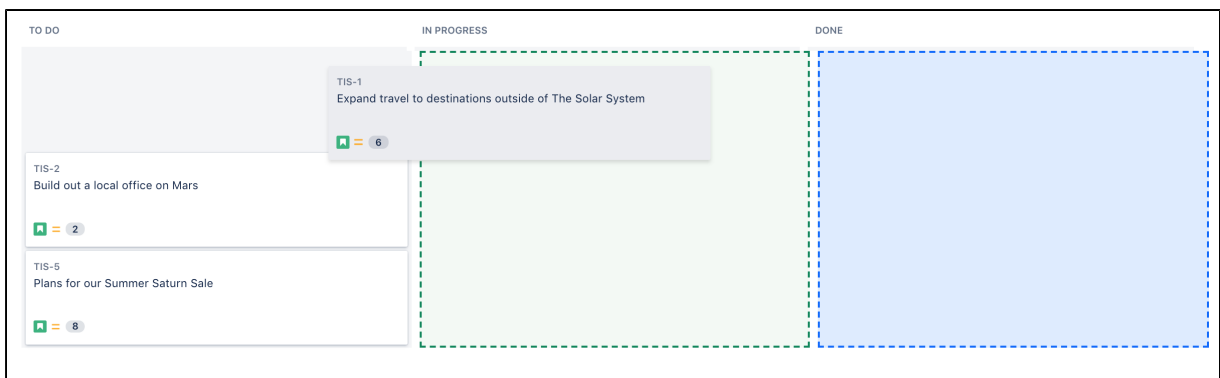
Work on an issue

By moving an issue to In Progress, you are indicating that work has started on it. At this stage, you will need to assign it to yourself, so that everyone knows who is working on it. You may also want to add some technical details about the issue – it's helpful for your teammates to know as much information as possible about the issues you're working on.

1. In the **Active sprints** of your board, select the issue in the **In Progress** column. The issue is displayed in the Issue Detail view.
2. In the Issue Detail view, select **Edit** from the 'cog' drop-down. The 'Edit Issue' dialog will be displayed.
3. If the issue is not assigned to you yet, type your name in the **Assignee** field.
4. Add some technical details about the issue as necessary.
   Note, you may want to specify a component or a version (if components and versions have been configured for your project), or add attachments as needed, etc.
5. When you're done adding technical details, click the **Update** button.

Collaborate with your team

While you're working on an issue, you will probably need to share some information, clarify some requirements, or discuss some details about the issue with your team. This is easy to do with issue comments.

1. Select the issue to add a comment.

2. Start typing your message in the **Add a comment...** footer or press the **m** key to jump straight there.
3. To email other users about your comment, simply mention these users in the **Comment** text box (by typing @User's Name). An email will be sent to the users' email addresses that are registered with their Jira accounts.
4. Select **Add** to save the comment.

Learn more about commenting in Jira

Create a branch

After assigning an issue to yourself and entering technical details about the issue, you're ready to do some coding. It is recommended that you create a branch when you start working on an issue. This way, you'll have your own stream of work that won't interfere with the rest of the team's work. It also ensures that your changes get reviewed before being merged back into the master branch.

Jira Software can be linked to a code hosting or repository management source, like Bitbucket Cloud or Bitbucket Server. This lets you open, collaborate, and manage source code directly from within Jira. Integrating an application lets you and your team create a branch directly from an issue, giving you a faster workflow from selecting an issue to coding.

Before creating a branch, you should already know how to use branches in the source repository that your team is using (Bitbucket Cloud instructions, Bitbucket Server instructions).

1. Select the issue for which you want to create a branch.
2. In the Issue Detail view, locate the **Development** panel, and click **Create branch**. The Create branch dialog will be displayed.
3. Include the issue key in the branch name. If you have workflow triggers configured, the status of the issue may automatically transition to 'In Progress'. See Configuring workflow triggers for more information.
4. Enter other details for the branch as necessary.
5. Click the **Create branch** button. Your branch will be created in your source repository.

Start coding

Now that you have created a new branch, you can start coding without worrying about your changes affecting the master branch. Ensure that you add the issue key that references the issue that you're working on. When you reference your issue key(s) in your development work, your connected development tools will also have links back to the relevant issues.

1. Go to your source repository where your new branch is created, and access your new branch.
2. In your new branch, implement your code or fixes as necessary.

Send your work for review

Use pull requests to tell your teammates about changes you've pushed to the repository. Once a pull request is sent, your team can review, discuss changes, or push follow-up commits.

1. Click **branches** in the Development panel to open a dialog in Jira Software that shows linked branches in the Software Configuration Manager (SCM). If Jira Software has been linked to more than one SCM, a tab will show for each SCM application (e.g. Bitbucket Server). The branches will be grouped under each SCM in these tabs.

TIS-45: 2 branches			
FishEye / Crucible 1	Stash 1		
Repository	**Branch**	**Pull request**	**Action**
Apollo UI	feature/TIS-45-email-non-registered-users-to-sign	MERGED	Create pull request
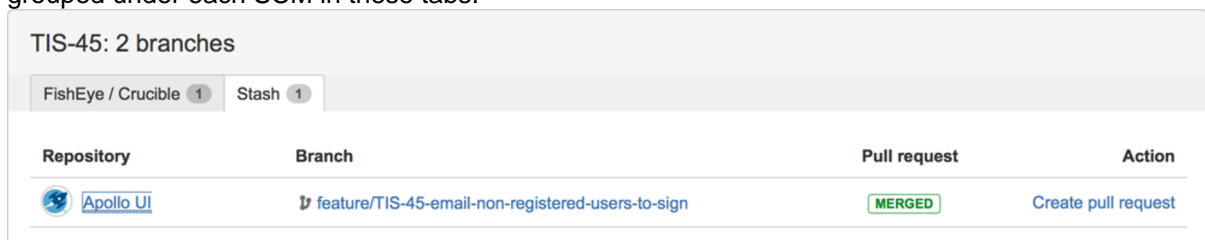
2. Click a repository or branch to open the linked SCM at the relevant repository or branch.
3. Hover over a **Pull request** status (e.g. MERGED) to show a popup displaying a link to the pull request.
4. Click **Create pull request** to create a pull request for the branch (to merge it back to master).

5. Include the issue key in the title of the pull request. If you have workflow triggers configured, the status of the issue may automatically transition to 'In Review'. See Configuring workflow triggers for more information.

Finish your work

After your work has been reviewed and merged with the master branch, the last step is to close or resolve the issue.

1. Locate your card in the **Active sprints** of your board.
2. Drag and drop the card to the **Done** column. If you have workflow triggers configured, the status of the issue may automatically transition to 'Closed'. You can skip the next steps. See Configuring workflow triggers for more information.
3. Select the **Resolution** for the issue from the drop-down menu.

4. Provide details as necessary.
5. Select **Close Issue**.

Great job! You have successfully gone through a simple workflow using Jira Software!

To do more awesome stuff, like managing your issues and customizing Jira Software, click the 'Advanced info' button below.

Advanced info

Learn to plan and estimate for scrum teams

Estimating and planning for agile projects can be really challenging regardless of how experienced you are. There's always competing priorities, resourcing issues, business demands and time constraints.

This topic can't make any of those things go away, but if you're a newish to agile scrum or just new to Jira Software, it will give you an overview of the Jira Software features to help you plan and estimate.

Learning objectives

The aim is to give you a good introduction to:

- Planning concepts and functions
- Estimating methodologies
- Analysis tools and techniques
- Optimization approaches

We've also asked some of our Atlassian teams to provide some tips and advice along the way.

Learning activities

There's optional activities outlined at the bottom of each topic. You don't have to do these, but it does help to put the theory into practice. You might want to create a dummy project to do this.

Pre-requisites

You need:

- Access to Jira Software. If you don't have a working copy, trial it here.
- The ability to create projects and configure boards.
  If you are working with an existing instance of Jira Software, you need this access to do some activities. If you create a dummy project, you will get this access by default.

Let's go

Learn more about agile

It helps if you're familiar with the agile language and concepts we'll use. So if this is a bit new to you or you feel a bit rusty, check out the Agile Coach, which covers lots of the basics.

Plan for the team

1. Plan for the team
2. Customize the team board
3. Estimate in story points
4. Analyze team reports
5. Optimize future plans

They say the best laid plans always go awry, and sometimes they do, but you still have to make them. At least in Jira Software, the plans you make are highly flexible, customizable, and configurable.

Start with a backlog

All agile projects start with a backlog of issues that represent the work that needs to be done: user stories, tasks, bug fixes, etc. Anything that's been determined is a discrete piece of work.

- Backlogs are all about priorities and doing the most important things first
- Use quick filters to just see the issues you want
- Create new issues and start sprints in the backlog

To get to your backlog, open your board and click **Backlog**. Control board details shown using the options in the Board menu.

You can add issues quickly by clicking **+ Create issue** below the backlog list.

> *"You're just setting yourself up for failure if you try to plan sprints too far ahead. You'll waste energy making decisions about things that will almost certainly change scope or priority in a few weeks. The backlog is always waiting to be re-prioritized"*
>
> *~ Atlassian Team Lead*
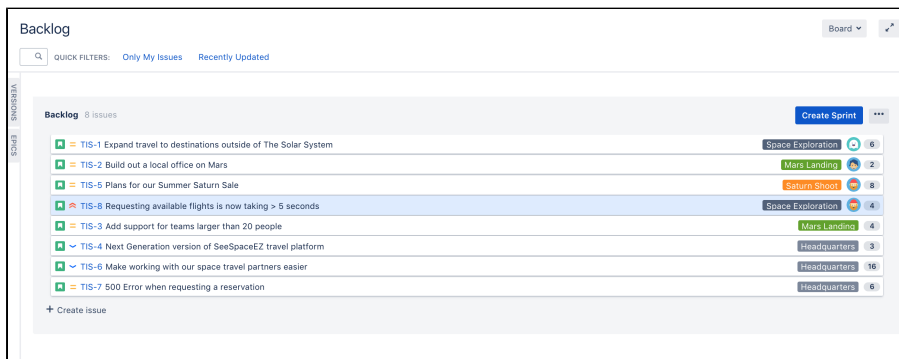
Use epics for context

Epics are designed to tell the bigger story. They can be used to capture broader work themes, such as 'performance' or 'user interface', or to track larger items of work. There are advantages to using epics.

- Epics can span multiple projects, which is especially useful if your board is set up to show issues from multiple projects.
- You can link an epic to a Confluence page to provide more information, such as design specs, business case, requirements, etc.

To create an epic

1. In your board backlog, click **EPICS** on the left side of the board, then click **Create epic**.
2. Complete the details of the epic in the window that appears and click **Create**.
3. Drag and drop backlog issues across to the epic they belong to.

40
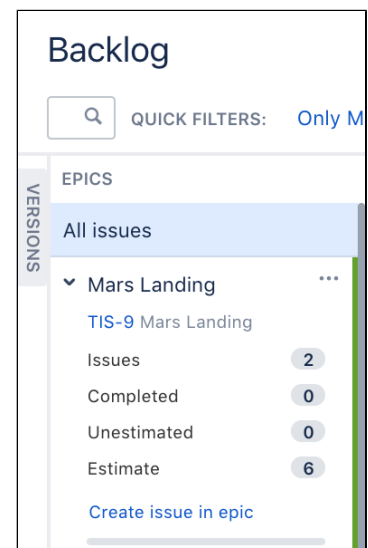
See Working with epics for more information.

Versions keep the release in order

Lots of teams release versions of their product as updates or feature sets for customers. Allocating issues to versions can help the team plan sprint work in the lead up to a release.

To create a version

1. Click **VERSIONS** on the left side of the board, then click **Create version**.
2. Complete the details and click **Create**.

You can then drag and drop backlog issues across to the version they belong to.

See Configuring versions in a Scrum project for more information.

Rank and prioritize

When your backlog issues are listed in order of priority, it's far easier to create sprints that are based on the work that needs to be done next.

Ranking is easily done by dragging and dropping issues in the backlog.

Rinse and repeat

41
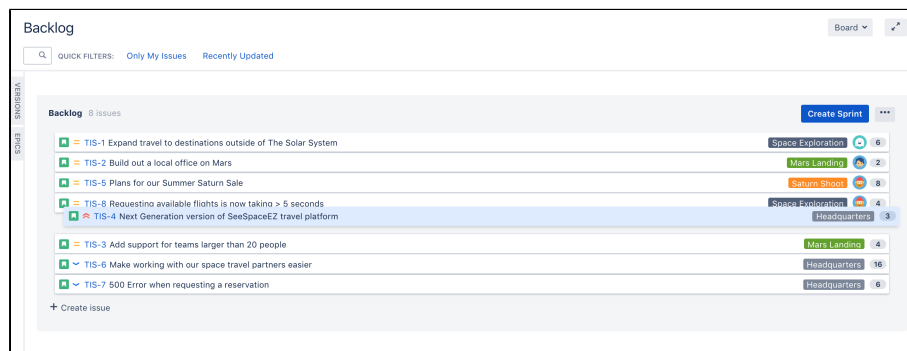
The best thing about scrum projects is that as you progress through sprints, the more clearly you can see what your team is capable of.

There's a heap of reports and charts that you can consult to help with understanding your team's productivity rates. We'll talk about how some of these can help you get better at planning a bit further along.

> **LEARNING ACTIVITY**
>
> Create a dummy project and choose **Scrum software development** as the project type.
>
> 1. Create a backlog of issues. You just need a title to add new ones.
> 2. Re-prioritize the backlog rank by dragging and dropping
> 3. Create an epic and add a few issues to it
> 4. Create a version and add a few issues to it

Customize the team board

1. [Plan for the team](#)
2. Customize the team board
3. Estimate in story points
4. Analyze team reports
5. Optimize future plans

The board is where all the action happens so you want to make sure that it's set up the way your team needs. You can change a heap of settings in the board configuration screens. Here's just a few.

⚠ You need to have special Board Admin rights to change board configurations. If you created the board, you already have them.

CONFIGURATION

General

Columns

Swimlanes

Quick Filters

Card colours

Card layout

Working days

Issue Detail View
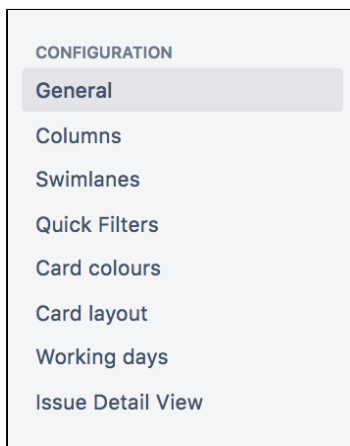
Board customization

Control everything from the board configuration page. In your board, go to **Board > Configure**.

- **Columns**
  Add statuses and change the workflow to suit your team structure. For instance if you want to show testing progress separate to dev progress, add a column. You can also switch on an indicator to help identify slow moving issues.
- **Card layout**
  Show up to three extra fields on the card layout. Say, if you want to show labels or components.
- **Card colors**
  If your team is 'visual' you can color code cards according to issue type, assignee, priorities, or queries

See Configuring a board to learn about the other things you might need.

Quick filters

Set up quick filters so you can just click once to show specific issues. You can set up a filter for issue types, assignees, unassigned, etc. Pretty much anything you need.

Go to **Board > Configure > Quick Filters**.

Filters work based on JQL (Jira Query language) which are really simple search statements. The result is something like:

✅ TIP: as you start typing in the JQL field, the syntax helper pops up to assist

> *"We just couldn't work as a global team without quick filters. The guys in San Fran and Gdansk need to be able to see exactly the same set of issues as us when we're discussing work online."*
>
> *~ Atlassian Dev Manager*

Quick Filters

Quick Filters can be used to further filter the issues in the board based on the additional JQL query.

Name	JQL	Description	
Engineering	✅ component = Engineering	All issues related to vehicle engineering	Add
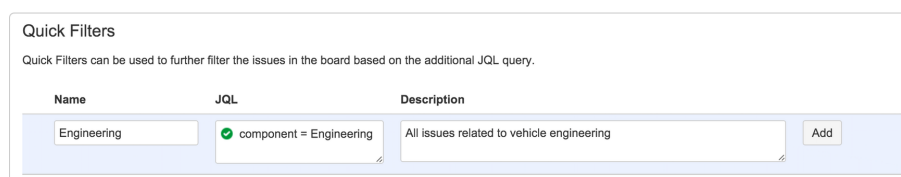
If you have a giant backlog, you probably use the issue search pretty regularly to find the issues you want. And if you're repeating searches using similar criteria, then other people probably are too.

Next time you are searching for a set of issues you know you will need again, save your search as a filter. Once you save a filter, you can share it, favorite it, get emails of search results, use it with a dashboard, and heaps more.

1. Go to **Issues > Search for issues**.
2. Enter your search filters and click **Save As**.

LEARNING ACTIVITY

1. Create a new quick filter and use it on your board.
2. Search for an issue and then save the search as a favorite.

Estimate in story points

1. Plan for the team
2. Customize the team board
3. Estimate in story points
4. Analyze team reports
5. Optimize future plans

There's a huge variety of ways to estimate stories, but the objective of every approach is to be able to get better at predicting how much work the team can do each sprint. In agile scrum, this translates to knowing the team's velocity.

Velocity measures the number of 'estimation units' that a team usually completes from sprint to sprint. It is effectively a productivity rate based on an estimation of volume of work, and it is best worked out in a measure other than 'time'.

Story points are a commonly used measure for estimating the size of an issue in scrum teams. During a typical planning session, a trivial bug fix might be estimated as a 1 or 2, and a larger feature might be anything up to a 12. Note that the scale of points does vary between scrum teams. Some use 1-12, others 1-5. The key is to use the same scale so that your velocity is consistently calculated.

If issues are estimated larger than this, they might need to broken into smaller stories or subtasks.

The focus is on story points here because it's the more common scrum estimation method, and we use it at Atlassian. Your company may use hours or ideal days. If you do use time for estimation, you might want to have a look at Configuring estimation and tracking.

> *"Even seasoned devs can't estimate accurately in hours/days. We just need to predict what can realistically get done with a reasonable degree of certainty. Story points all the way."*
>
> *~ Atlassian Developer*

A: It is much more useful in the long term to roughly predict how much work your team can do each sprint, than it is to try to estimate how long each story will take in time.

Story points enable the team to estimate stories in comparison to other stories, instead of forcing them to determine the time it will take to complete each story. Velocity is then worked out based on how many points the team can complete in each sprint. After a few sprints, team velocity stabilizes and estimation accuracy increases (or rather the same degree of inaccuracy is applied).

The problem with using hours to estimate is that the team is likely to estimate using the 'working hours' of the sprint (with some buffer time) and this is almost never an accurate reflection of a story's complexity or size. Accurate velocity is difficult to establish when stories are estimated in isolation, instead of by comparison.

Estimation, time and velocity are really critical things to understand. Read more in the Estimating an issue topic.

Estimate in points, track in time

Even when you estimate in story points, you can still track in time if you want. Knowing the team velocity (regardless of the measure used) enables you to roughly guess how long estimated backlog items will take to complete.

But Jira Software does also have a couple of dedicated fields (**Remaining Estimate** and **Time Spent**) to track time while using story points.

⚠️ NOTE: You need to be a board admin to make any configuration changes to a board. If you created the board, then you're already the admin.

Go to **Board > Configure > Estimation**.

Estimation

Issues can be estimated when in the Backlog to get an idea of how much work is being committed to in a sprint. Read more about estimation and tracking.

Estimation Statistic	Story Points ⇕

Estimate issues in the Backlog by entering values for **Story Points**. Your velocity from sprint to sprint will be measured against these estimates.

Time Tracking   ● None

Issues will burn down their **Story Points** value upon completion.

○ Remaining Estimate and Time Spent

Track time against issues using JIRA's **Remaining Estimate** and **Time Spent** fields.

- Select the **Estimation Statistic** (unit of estimation) - choose from story points, original time estimate, and issue count.
- Switch on the **Remaining Estimate** and **Time Spent** option to get a more accurate picture of how things are tracking in time units.
- If you leave **Time Tracking** as **None,** you can still refer to reports such as the Burndown chart to monitor progress.

Find out more about how time tracking work in projects in the configuring estimation and tracking topic.

The goal of velocity is to be able to look at a backlog of not particularly well-understood stories, and guess how many sprints it will take to complete. It requires a similar level of uncertainty for all of the estimates in the backlog. Determining accurate velocity relies on the equality of the inaccuracy, which is why you should NEVER re-estimate issues after a sprint has started.

LEARNING ACTIVITY

In your dummy project, go to **Board > Configure > Estimate** page and change the **Estimation statistic** to one of the following:

- ○ Story points
- ○ Original time
- ○ Issue count

Note the differences when you create and estimate an issue.

Previous   Next

Analyze team reports

1.
2.
3.
4. Analyze team reports
5. Optimize future plans

Jira Software comes with a whole lot of reports that will help everyone in the team understand where things are at. Here are just a few.

Access them all from the left navigation pane.

Velocity for the team

Teams want to know what their velocity is. They want to know how accurately they are estimating and how much they are getting done each sprint, in order to estimate better in future.

Velocity Chart

- See if the team is over-committing or under-committing stories in sprint planning
- Check for outliers and drill into a sprint to learn from past mistakes (e.g. a grossly underestimated story)
- Monitor how velocity alters during periods of team change or growth

Burndown for the leads

Team leads (Dev leads, Tech leads, etc.) usually spend half their time thinking ahead to future work, and the other half firmly involved in the current sprint. They need to know how things are going now to help steer the team to a successful sprint completion.

Burndown Chart

- If the **Remaining Value** line it is headed straight across and not down, you need to find out why
- A successful sprint is where the **Remaining Values** line hits the bottom on or before the **Guideline**

> *"Burndown is the fastest way for me to see what's going on and help out the team if things are looking a bit shaky."*
>
> **~ Atlassian team lead**

- If the **Remaining Value** line ends before the guideline, you may have a bunch of overachievers or under-committers on your hands!

Reports: Burndown Chart ·

Burndown Chart   Spring Cleanup   Story Points ▾   ⑦ How to read this chart
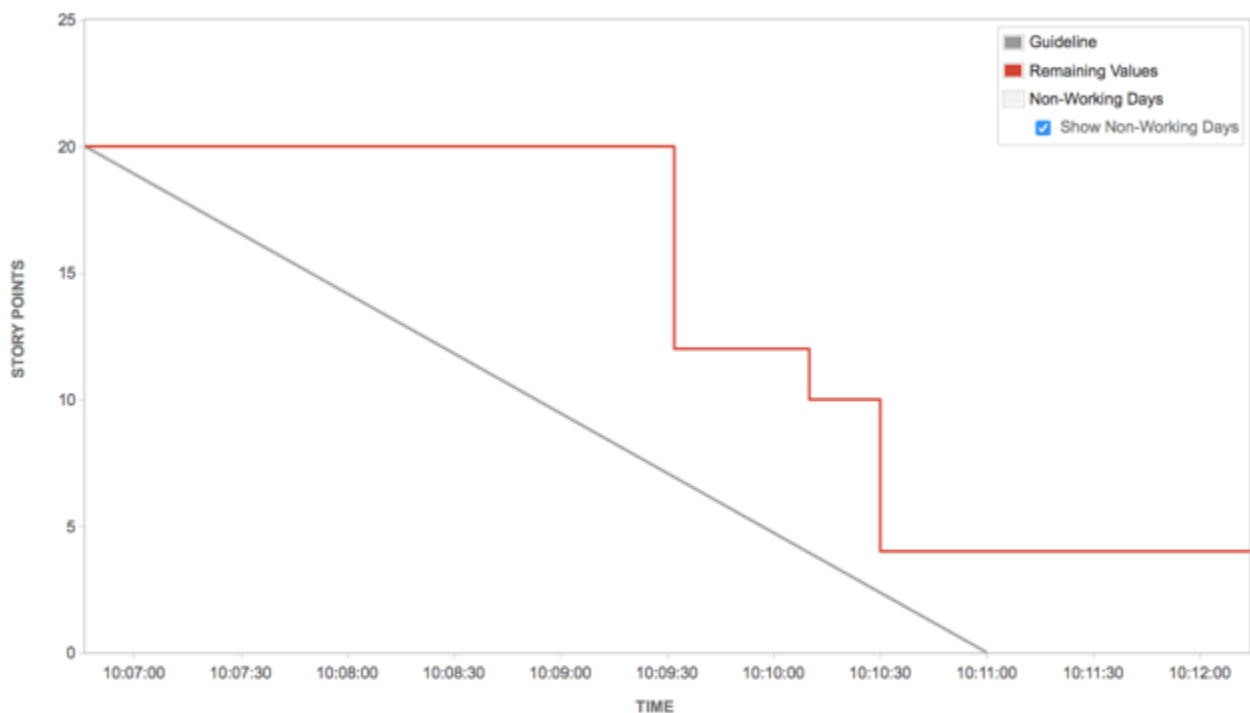
Date	Issue	Event Type	Event Detail	Inc.	Dec.	Remaining
01/Jul/15 10:06 AM	TIS-1	Sprint start		6		
	TIS-2			2		
	TIS-5			8		
	TIS-8			4		
						20
01/Jul/15 10:09 AM	TIS-5	Burndown	Issue completed		8	12
01/Jul/15 10:10 AM	TIS-2	Burndown	Issue completed		2	10
	TIS-1	Burndown	Issue completed		6	4

Burnup for the leads

Team leads (Dev leads, Tech leads, etc.) need to know how things are going now to help steer the team to a successful sprint completion as well as run retrospectives to identify the possible difficulties and bottlenecks that can be mitigated in the future.

Burnup Chart:

- If the Work scope line is spiking something has been added to the milestone. This might affect successful sprint completion.
- If the Completed work line does not spike it means that the team has problems completing stories. It might mean that work is more complicated than estimated.

Epics for product managers

Product managers (or project managers, depending on your structure) like to keep an eye on the big picture stuff, so they can check if project milestones are being hit, where dev effort is being spent, and if the backlog is in need of a re-prioritization. The Epic Report is a good read for satisfying some of these demands.

Epic Report

- Easily see the how near or far an epic is from completion. Check with the team if things look unbalanced.
- Monitor unestimated issues. Chat to the dev lead to see if anything needs further clarification or breakdown.

49

Epic Report ▾   Board ▾   ⌃

ⓘ **How to read this chart**

Understand the progress towards completing an epic over time. This helps you manage your team's
progress by tracking the remaining incomplete/unestimated work.

Hide this information

Bugfix and warranty issues ▾

TIS-100 Bugfix and warranty issues   Linked pages

Summary

Issues

Total: 32

Completed: 25

Unestimated: 3

Story Points

Total: 65

Done: 58

View Bugfix and warranty issues in
Issue Navigator

Status Report

Completed Issues   View in Issue Navigator

Key	Summary	Issue Type	Priority	Status	Story Points (58)
TIS-81	Afterburner initial design	🔲 Bug	↑ Critical	RESOLVED	-
TIS-82	Afterburner initial prototype	🔲 Bug	↓ Minor	RESOLVED	-

---

LEARNING ACTIVITY

You need to have historical sprint data in your project to view the above reports. If you don't have access
to an existing project, just browse the report help topics:

- Sprint Report
- Burndown Chart
- Epic Report

Previous   Next

Optimize future plans

When things are running well, velocity is stable, the team is feeling challenged and even the product manager is happy, it is not time to stop planning.

Plan ahead

Everything learned from burndowns, reports, story estimation, sprints and standups is ready and waiting to help make planning better for future sprints and issues. Even if the team changes, the budget is cut, or the company you work for insists on a whole new focus, the data collected is still really valuable.

Here's the top things you might change next time, based on what you learn this time:

A high performing team that always completes a sprint with no work to spare may need to be challenged with more points in the next sprint. But be careful of tipping the scale too far in the other direction. Teams that rarely complete a sprint without leftover points may feel like they are underperforming, even if they aren't. When things are going well, you don't want to mess with the formula. But when things aren't going as well, it can help to change things around a bit (if you have the luxury to do so). This could take the form of swapping responsibilities for a sprint, or introducing a different kind of sprint, such as an innovation sprint or bugfix only sprint. Change can be re-invigorating, and it can still contain work!
Failure to meet committed story points can be disappointing for a team, especially if they don't know why. Check in with the PM or person who reported the issue to clarify details during estimation, and raise concerns during retros. If estimates are way off due to lack of basic information, for example, this needs to be addressed.

> "One of the best things about estimating as a team is that it sparks some really important discussions about how we build our product and how we can get better at what we do."
>
> **~ Atlassian Product Manager**

Don't forget the invisible things

Scrum is all about team and a lot of things happen during sprints that need to be captured and fed into future plans. Here's a few things that new starters should know and experienced operators sometimes forget.

- **Always hold a retro. Always.**
  Retrospectives can give you insights that reports can't. Make sure you discuss any issues that come up, which might result in adjustments to planning. Record insights (preferably in Confluence) and brag loudly about the good stuff.
- **Never assume the team knows how they are performing**
  Just because velocity reports are available and are viewed in monthly team meetings, doesn't mean the team knows how they are going. Scrum leads and team leads need to use words too. Who doesn't like being told in person they are going well.
- **Say it with meaning even if you've said it before**
  Standups and planning can become a bit routine over time, especially with experienced teams who have worked together for ages. Try not to let the rituals become lip service, make sure people are saying meaningful things and are listening to each other actively.

> **FURTHER LEARNING**
>
> If you want to learn more about estimating and planning, check out some of these options:
>
> - Browse the Jira Software documentation
> - Take a few free classes at the Atlassian University
> - Search and ask the Atlassian user community for answers
> - See if The Agile Coach can help
> - Search all Atlassian documentation

Previous

Installing Jira Software

Jira Software can be customized to suit your needs. You can also extend it and link it with other applications to provide you with the perfect solution to track all your work in one place.

You can install Jira Software Server anytime.

Before you start, read the Jira Software release notes for the version that you are installing or upgrading to, then follow the instructions below.

1. Download the installer here.
2. Once the installer has downloaded, run it and follow the Installing Jira applications steps.

If you experience any problems with your installation or you have any questions, contact Support.

⊘ **Get more out of your new JIRA Software instance**

Connecting Jira Software to other Atlassian products enables a host of new integration features. Read the pages below to learn more:

- Using Jira applications with 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.

- Using Jira applications with Portfolio for Jira — Portfolio for Jira provides a single, accurate view for planning and managing initiatives across multiple teams and projects with ease.

Using Jira applications with Advanced Roadmaps for Jira

Advanced Roadmaps for Jira (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 from Jira
- 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
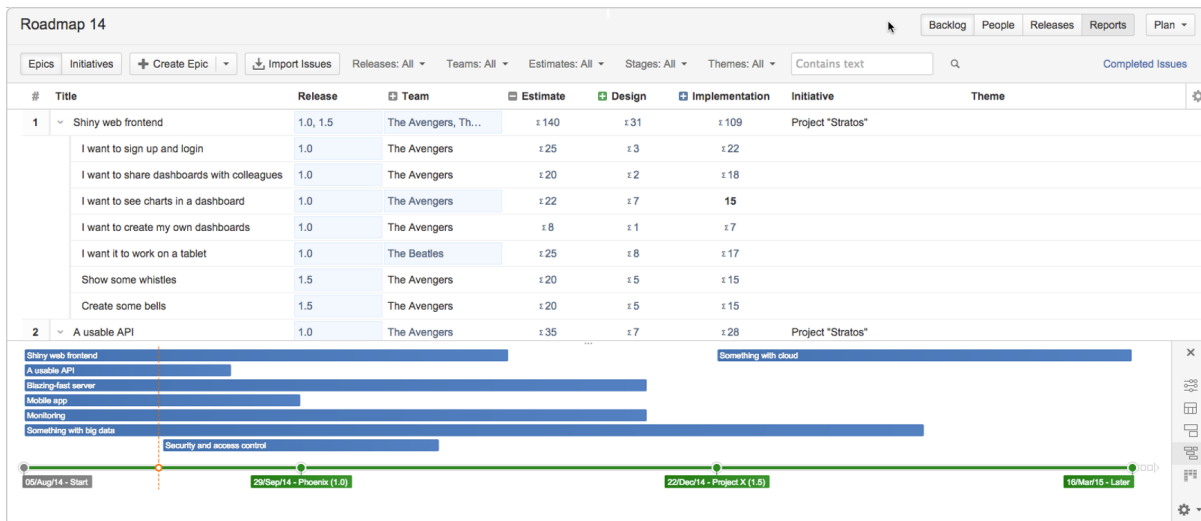
Plan automatically

- 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 member availability, including time off, training and inter-team commitments
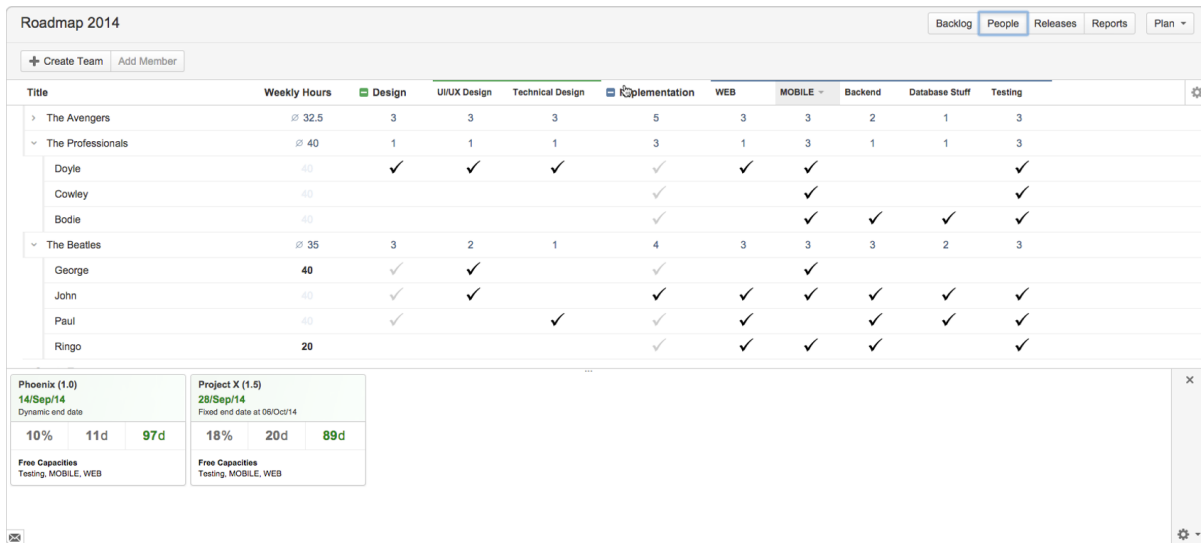
Keep up to date

- Keep the long term plan in sync with reality by adjusting delivery 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
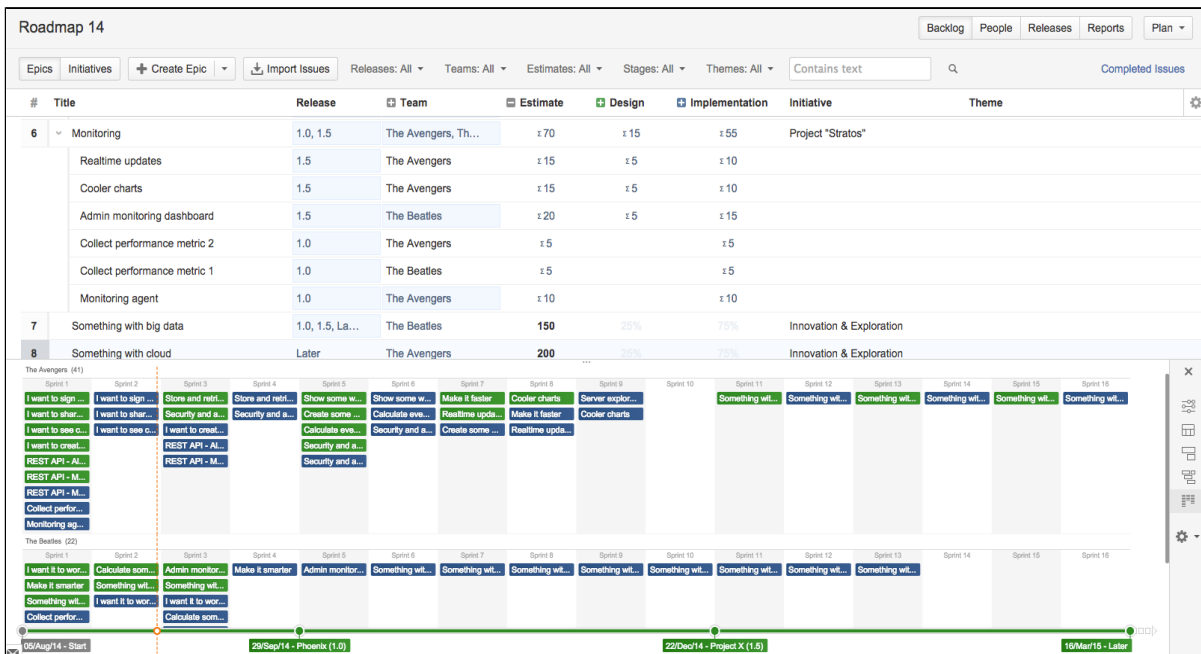
Make realistic commitments

- 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
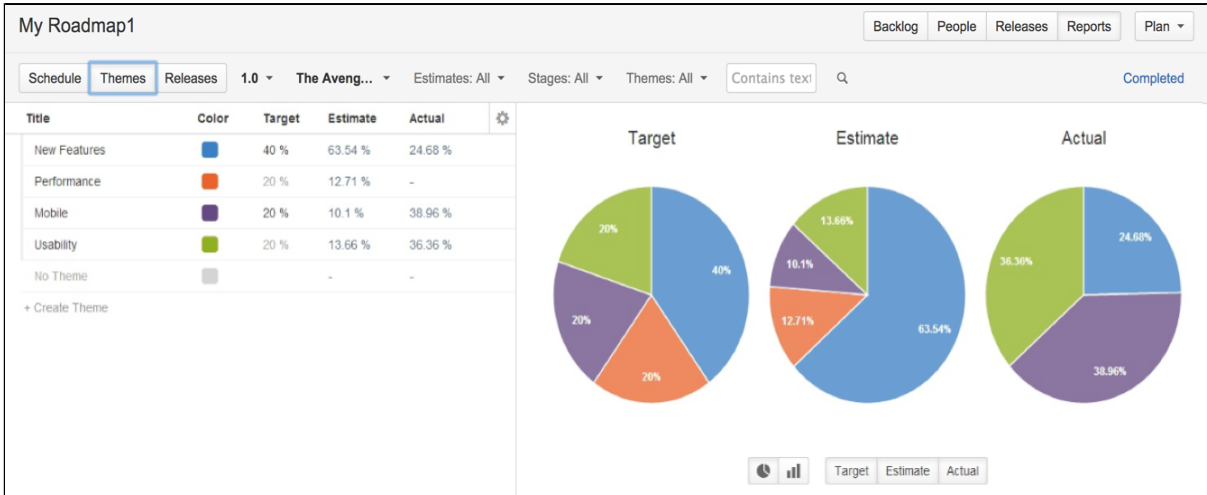
Model changes

- 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 affecting the underlying data in your Jira projects

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

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.
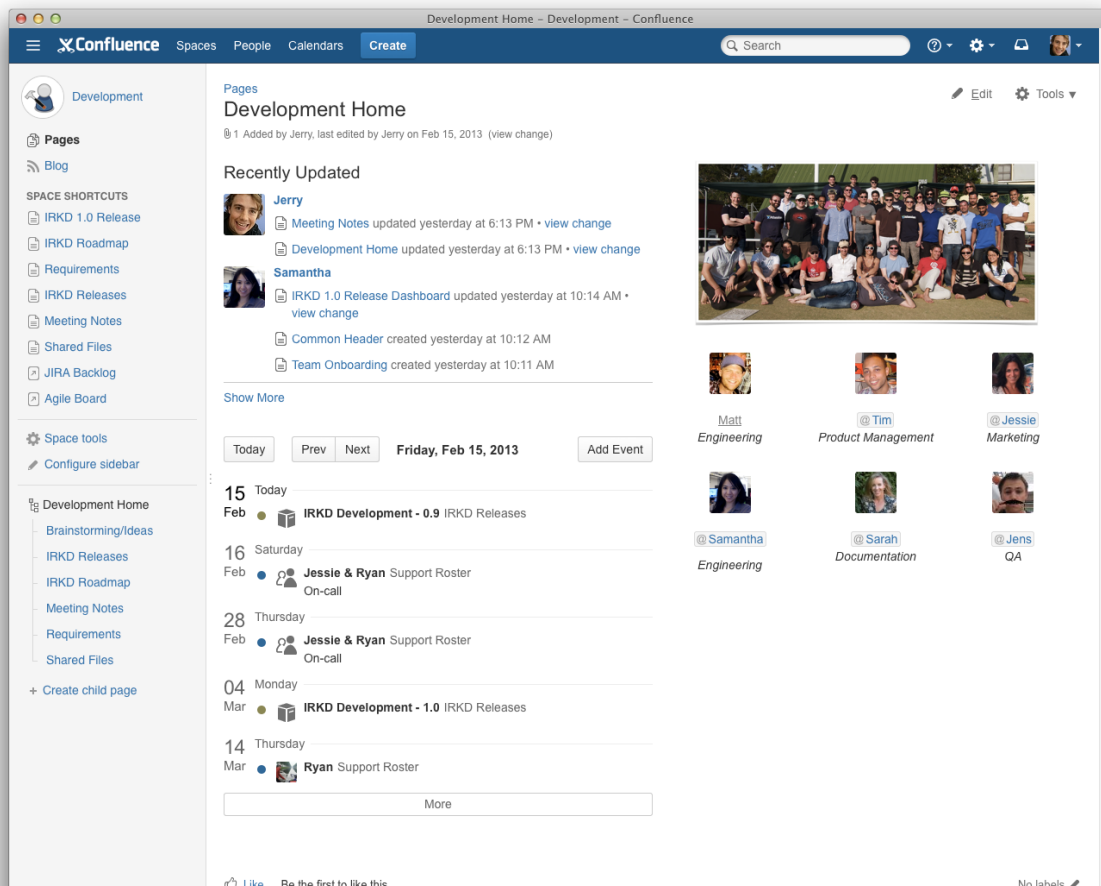
Why use Confluence with Jira?

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.
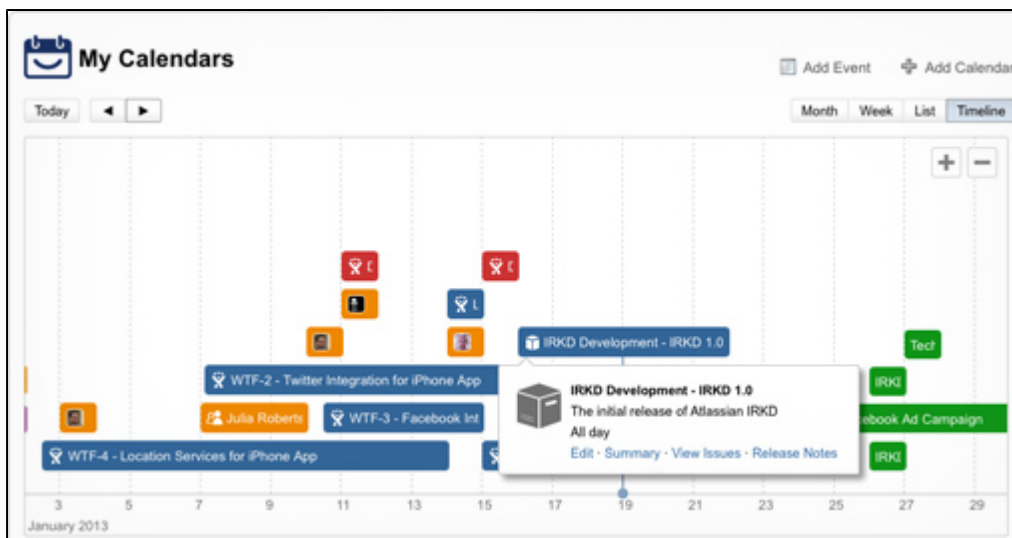
To install this feature, please visit Atlassian Marketplace.

Insert issues on any Confluence page using the Jira Issues macro

58

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.

Leading an agile project

You may be the iteration manager, product owner, agile coach, or team lead for your team — if you are responsible for running an agile project with Jira Software, you are in the right place! The topics in this chapter will help you configure your existing Jira Software instance to suit your agile development process.

In the wrong place? See the following chapters instead:

- *If you are a team member working with Jira Software, see Working in an agile project.*
- *If you are an administrator installing or configuring the Jira Software server, see Administering Jira Software.*

Where do I start?

- If this is the first time you're using Jira Software, read our Getting Started guide before you start reading other topics.
- If you're already familiar with Jira Software, start with Starting a new project to set up your project, or use the search box below to find the topic you need.

Search the topics in 'Leading an agile project':

Overview

The topics in this chapter cover the activities that you will need to do in each stage of the agile development lifecycle:

Starting a new project
Time to get everything set up! This is the stage where you create a new Jira Software project, configure your development tools (e.g. create repositories in Bitbucket, create projects in Bamboo, etc), and determine how you will manage documentation and team calendars.

Learn more...

Building a backlog
A backlog is a list of outstanding work. Before starting new work, you will build a new backlog or groom an existing backlog. This involves creating, ranking, and estimating issues, so that you have a prioritized list of tasks to work from.

Learn more...

Planning a version
Ready to start work on a new version? To plan a version, you need to set up the version in Jira Software and configure your development tools appropriately (e.g. branch your repository, set up Bamboo plans, etc).

Learn more...

Getting to work
All systems are go! Once work begins, you will be running sprints and tracking your team's progress. For teams using Scrum, this involves planning and starting sprints, monitoring sprints via boards and charts, and completing sprints. For teams using Kanban, this involves monitoring the work in progress, and limiting work to the constraints for your team, via boards and charts.

Learn more...

Releasing a version
This is the culmination of your team's hard work. Before releasing a version, you would typically check if there is outstanding work for the version. If everything's good to go, you can then deploy the version, and create the related release reports and documentation.

Learn more...

Reporting

Reporting is an activity that occurs throughout a development cycle. You may need to make reports available to different stakeholders or use them yourself. Jira Software has a number of reports that you can use to provide different information about the progress of your team's work.

Learn more...

Starting a new project

It's time to start your project! You know what you need to do, and you have a team that is excited to start working. You want to get everyone off to the best start, so let's make sure that your systems are set up and ready to go.

The documentation in this section will help you configure Jira Software for a new agile development project. This includes creating a new Jira Software project, configuring a board (Scrum/Kanban), configuring your development tools to work with Jira Software, and more.

Search the topics in 'Starting a new project':

Overview

Note: it is assumed that you already have a Jira Software instance. If you don't have Jira Software, install Jira Software Server (Installing Jira applications) or sign up for a Jira Software Cloud site account.

Creating or configuring a Jira Software project	A *Jira Software* project is simply a collection of issues (stories, bugs, tasks, etc). You would typically use a project to represent the development work in Jira Software. This could be the work related to a product, a team, etc. Learn more: Configuring a project
Setting up your repositories, build plans, and review projects	The start of a project is where you would set up new repositories in Bitbucket or FishEye, create build plans in Bamboo, and perhaps create new review projects in Crucible (if you are not using Bitbucket). If you're using alternative tools, consult your vendor's documentation. Learn more: Create a repository (Bitbucket Cloud), Creating repositories (Bitbucket Server), Configuring plans (Bamboo), Creating a project (Crucible)
Connecting your development tools to Jira Software	Jira Software can be connected to a range of development tools to help you keep your project tracking in sync with your development work. For example, if you have Bitbucket connected to Jira Software, you can configure issues to automatically transition when commits are made. Learn more: Configuring development tools
Connecting your collaboration tools to Jira Software	Every software project requires collaboration, whether it be via chat rooms, online documentation, or team calendars. If you have HipChat (chat), Confluence (wiki) and/or Team Calendars for Confluence (calendars), you can connect them to Jira Software to help you plan and run your project more efficiently. For example, you could get issue notifications in your chat rooms, link specifications to epics or retrospective notes to sprints, include team events on your project calendar, and more. Learn more: Configuring collaboration tools

Already set up your project?

Next: Start building a backlog

Project lifecycle
NEW PROJECT BACKLOG NEW VERSION WORK RELEASE REPORTING
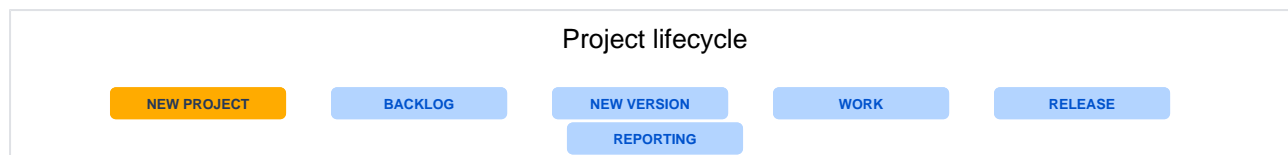
Configuring a project

Your issue tracker should be the hub of your development project. When properly configured, Jira Software helps your team members prioritize and organize work better, so they can spend more time developing great software instead of wrangling issues.

A project is simply a collection of issues (stories, bugs, tasks, etc). You would typically use a project to represent the development work for a product, project, or service in Jira Software.

Scrum or Kanban? *Scrum has an iteration-based approach, which is generally a good fit for teams developing products, particularly if your team is releasing new versions on a regular schedule. Kanban is better suited for a continuous flow of work (e.g. service-oriented teams), where its constraint-based approach helps prevent your team from being overloaded.*

Before you begin

Ideally, you should be a **Jira administrator** (i.e. someone with the 'Jira Administrators' global permission), if you want to set up a new project in Jira Software. Many of the tasks involved can only be performed by a JIRA administrator, such as creating a project, modifying a workflow, etc. For more information, see Permissions overview.

Overview

The topics in this section cover the activities that you will need to do when configuring a project:

Create a project	Your first task is to create a Software project and configure it according to the needs and requirements of your development team. You need to be a Jira administrator to create a project, but you can configure most project details (e.g. name, avatar, etc) if you are a project administrator.
	Learn more: Defining a project (Jira Admin documentation)
Configure the board for your project	A board is the central tool for working with issues in a Scrum or Kanban development project. A Scrum or Kanban board will be created when you create a Scrum or Kanban development project respectively. You can add more boards (of either type) to the project, if you like.
	Learn more: Configuring a board
Configure project permissions	You can control access to your project by configuring a permission scheme. A permission scheme maps users, user groups, roles, etc to the project functions (e.g. Assign issues). Your new project will be pre-configured with the default permission scheme.
	Learn more: Managing project role memberships
Tweak issue types, workflow, screens, and fields	Your project is pre-configured with issue types, a workflow, screens, and fields. You can change these to suit any development process. For example, you may want to create a 'Design' issue type or add a 'QA review' step to your workflow.
	Learn more: Defining issue type field values, Workflows, Defining a screen, JIRA custom fields (Jira Admin documentation)
Create components	Components can be used to group issues in a project. However, in an agile project, you should consider using epics to group issues (stories) instead, as there are more features that support working with epics, like the epic column on boards, reports, etc.
	Learn more: Managing components

Creating a board

You need a board so that you can view and work on issues in Jira Software. A board displays issues from one or more projects. You can either copy a board that someone else has created, or create a new board for yourself. You can create as many boards as you like.

There are two types of boards:

- **Scrum boards** are for teams that plan their work in sprints.
- **Kanban boards** are for teams that focus on managing and constraining their work-in-progress. Because work is not planned in advance using discrete time periods or versions, Kanban boards do not have a Backlog screen.

Learn more: What is a board?

Before you begin

Any user can create a board, but certain permissions are required to share the board with other users:

- If you create a board via **Boards** (in header) > **View All Boards**, you will not be able to share it, unless you have the 'Create Shared Objects' global permission.
- If you create a board via the following methods, you do not need the 'Create Shared Objects' global permission to share the board:
  - Creating a project (where a board is created for the project by default)
  - Setting up Jira Software for the first time (where you're prompted to create a project, which also creates a board for the project)
  - Copying a board (the copied board will be shared with the same users as the original board)

Creating a new board

1. Log in to Jira Software.
2. If you're setting up Jira Software for the first time, do either of the following actions. Otherwise, please continue.
   - *If you're creating either a **Scrum software development project** or a **Kanban software development project**, follow the prompts to create your project. A board is created for your project by default.*
   - *If you're creating a **Basic software development project**, follow the prompts to create your project. After your project is created, click **Create board** under your project name, and follow the prompts to create your board.*
3. Select **Boards** > **View All Boards** from the top navigation bar.
4. Select **Create board** at the top-right of the page, and choose whether to create a Scrum board or Kanban board, as described below. Note that you cannot change the board type after creation (that is, a Scrum board cannot become a Kanban board, and vice versa).

   a. Click **Create a Scrum board**, then choose whether to base your board on a new Software project ('Browse Projects' permission required) or an existing project.
   *Note, you can also choose to **Create a Scrum board with sample data** ('Browse Projects' permission required). This will base your board on a new project pre-populated with sample data.*
   b. Follow the prompts to set up your board and project (if you are basing your board on a new Software project). This will create a pre-configured Scrum board containing all the issues in your chosen project(s). Your Scrum board will have an issue filter with the following query:

   ```
   project = "[YOUR PROJECT(S)]" ORDER BY Rank ASC
   ```

a. Click **Create a Kanban board,** then choose whether to base your board on a new Software project ('Browse Projects' permission required) or an existing project.
*Note, you can also choose to **Create a Kanban board with sample data** ('Browse Projects' permission required). This will base your board on a new project pre-populated with sample data.*
b. Follow the prompts to set up your board and project (if you are basing your board on a new Software project). This will create a pre-configured Kanban board containing all the issues in your chosen project(s) that do not belong to a released version. Your Kanban board will have an issue filter with the following query:

```
project = "[YOUR PROJECT(S)]" AND (fixVersion in unreleasedVersions() OR fixVersion is EMPTY)
ORDER BY Rank ASC
```

Before you begin: You must have access to at least one saved issue filter (either your own filter, or one that someone else has shared with you). If you don't, first create and save a new issue filter (see the documentation on issue filters and JQL if you need help).

Click **Create a Kanban board** and base your board on an 'existing Saved Filter'. Follow the prompts to set up your board. Your new board will be available to all users who have access to your chosen filter. See the examples below for sample JQL for your board's filter.

Note, if you are the owner of the issue filter, you can edit it via the **Edit Filter** link. Otherwise, this link will not appear.

5. Your new board will be shown. At the top is a link that you may want to send to other people so that they can use your board.

✅ **Congratulations — you have created a new board!**

Copying an existing board

If you would like to create a board that is similar to one you are already using, you can simply create a copy.

Your new board will be based on the same issue filter as the original board. You will be the administrator of the new board, but not necessarily of the filter, so you may not be able to edit the filter. However, once your new board is created, you can easily choose a different filter (e.g. to view a different project) — see Configuring filters.

1. Select **Boards** > **View All Boards** from the top navigation bar.
2. The **All boards** page will display. Select the **Copy** link corresponding to the board of interest.

Alternatively, to quickly create a copy of the board you are currently viewing, simply select **Copy** from the **Board** menu at the top right of the page.

Deleting a board

To delete a board:

1. Select **Boards** > **View All Boards** from the top navigation bar.
2. On the **All boards** page, for the board that you want to delete, go to the three-dot menu and select **Delete**.
3. In the dialog window, select **Delete** again to confirm your action.

ⓘ If you're deleting a scrum board that contains active sprints, you'll see a notification about it in the dialog window. We recommend that you complete the active sprints before deleting the board. So, you'll save the teamwork done during the sprints and won't have to restore the data.

At the same time, you aren't blocked from deleting the board.

Example JQL queries for board filters

Here are a few common queries that you can use for your board's filter:

- Select all issues that are Unscheduled or in an Unreleased Fix Version:

```
project = GHS AND (fixVersion in unreleasedVersions() or fixVersion is empty
```

- Select all issues you are interested in:

```
(assignee = currentUser() or reporter = currentUser()) AND (fixVersion in unreleasedVersions() or
fixVersion is empty)
```

- Show all issues that you have participated in and have been updated in the last week. ⓘ This requires the JIRA Toolkit app.

```
updatedDate > -7d AND Participants = currentUser()
```

- Select all issues for a team (using a Label custom field named 'Team')

```
(team = ateam or team = dreamteam or team = engineroom) AND (fixVersion in unreleasedVersions()
or fixVersion is empty)
```

- Only select my bugs for a bugfix team

```
project = GHS AND team = bugfix AND issuetype = bug AND (fixVersion in unreleasedVersions() or
fixVersion is empty)
```

- Try *fixVersion = earliestUnreleasedVersion(PROJECT KEY)* to see all issues in the next Fix Version to be released:

```
fixVersion = earliestUnreleasedVersion(PROJECT KEY)
```

See JQL for more information on using JQL for your board. Let your imagination run wild!

Next steps

ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- To learn more about your new board, see What is a board?
- If you wish to make any changes to your new board, see Configuring a board.
- If you wish to make any changes to the issue filter for your board, see Configuring filters.
- To learn more about adding a column to your board, see Configuring columns.

What is a board?

A board displays issues from one or more projects, giving you a flexible way of viewing, managing, and reporting on work in progress. There are two types of boards in Jira Software:

- **Scrum board** — for teams that plan their work in sprints
- **Kanban board** — for teams that focus on managing and constraining their work-in-progress

You can use a board that someone else has created, or create your own — you can create as many boards as you like. A project can have multiple boards, and a combination of Scrum boards and Kanban boards, if you choose.

Accessing a board

- Accessing a board
- Using the issue detail view
- Using the project sidebar
- Accessing a cross-project board
- Next steps

1. Navigate to your desired board.
2. On the Project sidebar, click one of the following, depending on what you need to do:

- **Backlog**, **Active sprints**, or **Reports**, if you are working on a Scrum board
- **Kanban board**, **Backlog***, or **Reports**, if you are working on a Kanban board.

* Kanban boards don't have a separate Backlog view by default. The Backlog view will only be available if you enable the Kanban backlog.

On a Scrum board

Backlog

The **Backlog** of a Scrum board shows the issues for your project (s) grouped into a backlog and sprints. In the Scrum backlog, you can create and update issues, drag and drop issues to rank them, or assign them to sprints, epics, or versions, manage epics, and more. You would typically use the Scrum backlog when building a backlog of issues, planning a new version, and planning a sprint.

Active sprints

The Active sprints of a Scrum board displays the issues that your team is currently working on. You can create and update issues, and drag and drop issues to transition them through a workflow.

Reports

Reporting is an activity that you will be doing throughout a project. Jira Software has a range of reports that you can use to show information about your project, versions, epics, sprints, and issues.

On a Kanban board

Backlog

By default, Kanban boards have the Backlog column, where you plan work for your team. Planning work in a small column becomes difficult as more issues are added in the column. By enabling and using the *Kanban backlog* , you have a bigger space to create and rank issues, and select issues for your team to start working on.

Kanban board

The Kanban board is a board that was created using the "Kanban" preset (see Creating a board).

Kanban is based on the continuous delivery of work. Rather than plan iterations, the flow of work is constantly monitored to ensure that there are always tasks being worked on. This means that when tasks are completed, new tasks are pulled into work-in-progress.

Use the Kanban board if your team focuses on managing and constraining work-in-progress.

Reports

Reporting is an activity that you will be doing throughout a project. Jira Software has a range of reports that you can use to show information about your project, versions, epics, sprints, and issues.

Using the issue detail view

Whether you're working on a Scrum or Kanban board, you can click on any issue to view it in more detail.
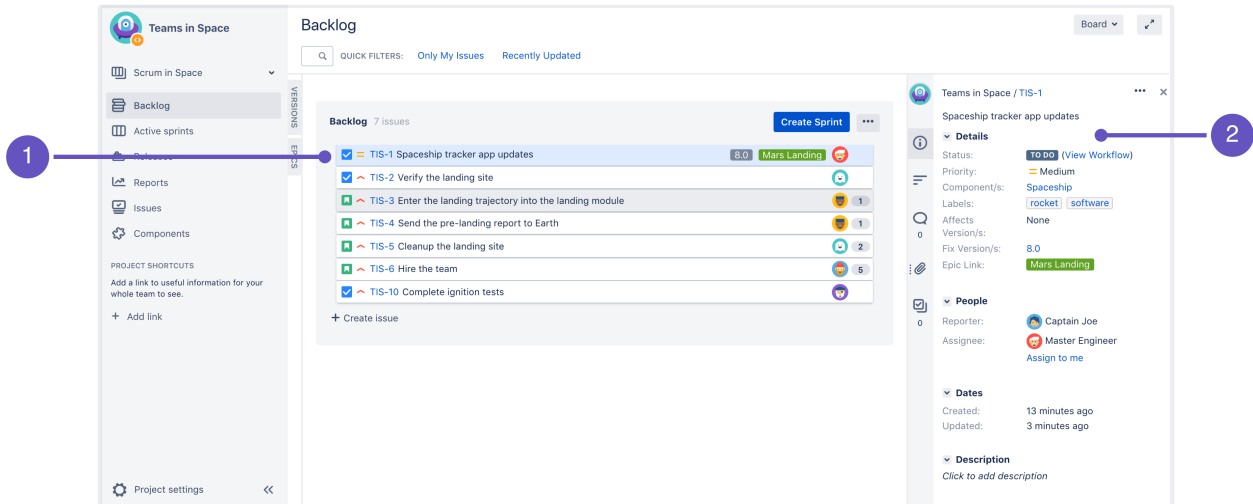
1. **Selected issue.**
2. **Issue detail view.**

Jira issue view can be configured to display various system and custom fields that describe the issue. You can customize this view by adding new fields, deleting fields, or changing their positions. For example, you may want to show an issue's Resolution, Priority, Labels, Security Level, etc. See Configuring the issue view for more information.

Using the project sidebar

The project sidebar is your one-stop shop for anything and everything that concerns your project. Any tab you click on the project sidebar opens a page that contains information for the project that you're currently working on. Let's say you're working on an agile project and you're currently on a Scrum board for that project. When you click **Backlog** on the project sidebar, it opens the Backlog page, which contains the issues, stories, epics, and other details of your project.
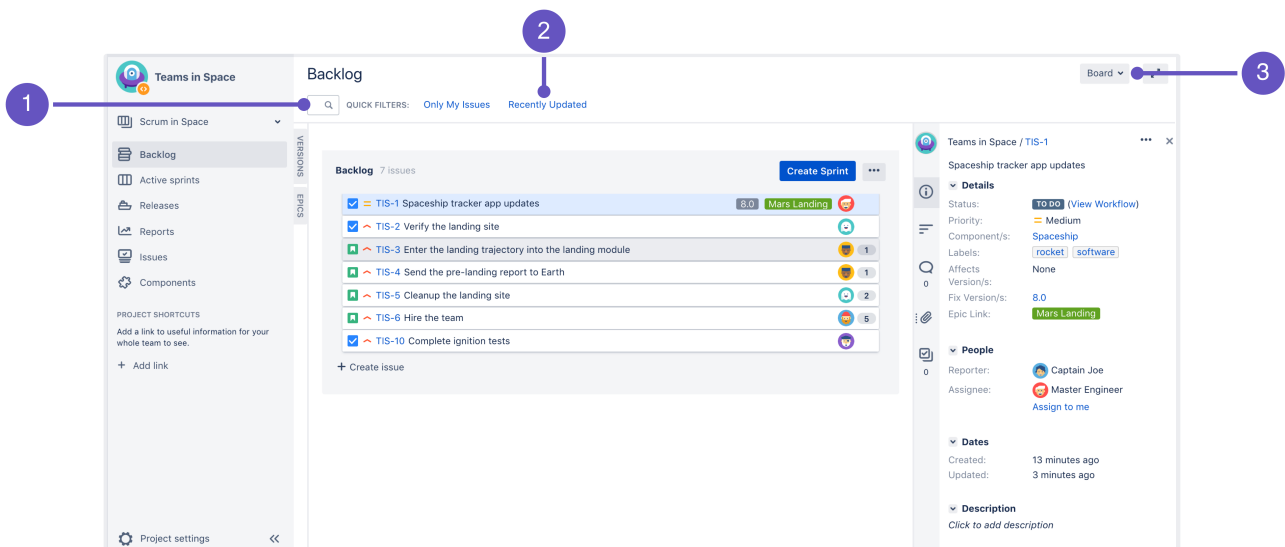
1. **Instant filter (Backlog only):** Type part of the issue summary (issue key, issue type, assignee) to search for matches to be displayed on the Backlog.
2. **Quick filters:** Choose any of the filters to quickly display only issues that are assigned to you, or issues that are recently updated.
3. **Board drop-down:** Create, configure, or copy the current board.

Accessing a cross-project board

A cross-project board is not specific to a single software project. That is, the board's filter may include a non-Jira Software project, or it may include multiple projects. How you navigate to a cross-project board, and the navigation options that you'll see on the project sidebar of the board are different, compared to a board with a single JIRA Software project.

There are two ways to navigate to a cross-project board:

From Projects in header

If you navigate to **Projects** (in the header), then select the cross-project board, the project sidebar will display project-centric options, like the **Issues** link and project shortcuts.

Screenshot: Project sidebar (navigating to the board via Projects menu)
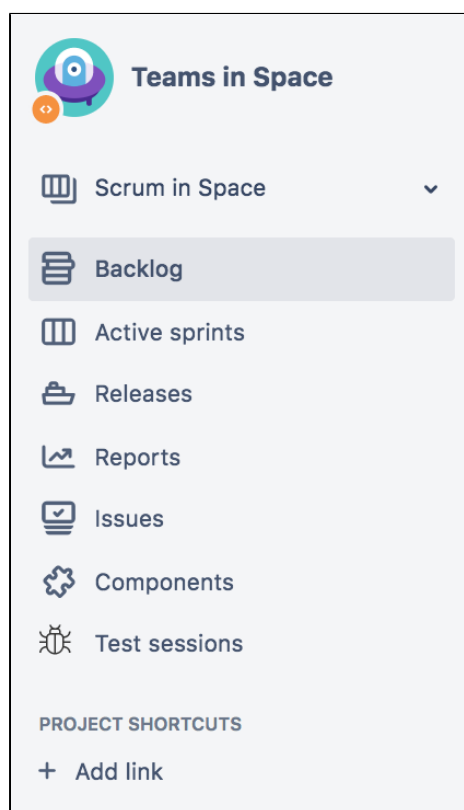
From Boards in header

If you navigate to **Boards** in the header, then select the cross-project board, the project sidebar will *not* display project-centric options, like the **Issues** link and project shortcuts.

This is because Jira Software cannot determine the project context for the cross-project board.

Screenshot: Project sidebar (navigating to the board via Boards menu)
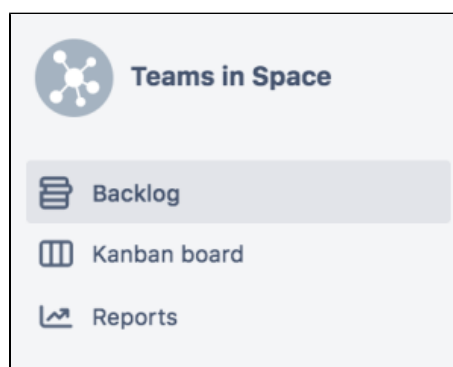
As a result, we recommend that you navigate to a cross-project board via the **Projects** menu.

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Creating a board
- Configuring a board
- Using plan mode

- Using active sprints
- Reporting

Configuring a board

Configuring a board allows you to edit the mapping of workflow statuses to columns of a board, as well as edit the columns, swimlanes, and quick filters of a board, and customize the card colors and displayed issue fields.

Before you begin

You must be a **Jira administrator** or a **board administrator** for the board to modify its configuration.

Accessing a board's configuration

1. Go to the desired board and select **Board** > **Configure**.
2. On the **Board Configuration** screen, select the desired tab (**Columns**, **Swimlanes**, etc).

Screenshot: the 'Board Configuration' screen of a Scrum board ('General' tab).

Screenshot: the 'Board Configuration' screen of a Kanban board ('General' tab).

Renaming a board

1. Go to the desired board and select **Board** > **Configure**.
2. On the **Board Configuration** screen, click the 'pencil' icon ✎ (which will appear when you hover) to edit the name of your board. Press the **Enter** key when finished.

Changing a board's administrators

A board's *administrator* is the only person who can change the configuration of a board along with Jira administrator users. By default, the administrator of a board includes the person who created it. If you are a board administrator or a Jira administrator, you can change the administrators of a board. See Per missions overview for more information.

1. Go to the desired board and select **Board** > **Configure**.
2. On the **Board Configuration** screen, click any of the name(s) in the **Administrators** field (a 'pencil' icon ✎ will appear when you hover over them).
3. Enter the names of the desired users or groups. If you start typing the name, a list of users and groups will be suggested. Press the **Enter** key when finished.
   Note, the suggested user list will not display if you don't have the 'Browse Users' global permission. However, you can still enter the user names and group names manually. See Permissions overview fo r more information.

You may also want to give the new administrators rights to manage the JIRA filter on which the board is based – see the Jira Admin documentation on Managing filters.

Sharing a board

A board is available to all users who can view the saved filter on which the board is based. (Note that they will also need 'Browse' project permission for the project(s) whose issues are shown on the board. See Permi ssions overview for more information.)

If you wish to share a board with different people, you will need to either edit the saved filter (see the documentation on issue filters), or choose a different filter (see Configuring filters).

Next steps

(i) **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following topics for more information about configuring your board:

- Configuring filters
- Enabling ranking
- Configuring columns
  - Using the Simplified Workflow
- Configuring swimlanes
- Configuring Quick Filters
- Customizing cards
- Configuring estimation and tracking
- Configuring the issue view
- Configuring working days

Configuring filters

Your board's filter is an issue filter (a JQL query) that specifies which issues are included on your board. For example, your board may include issues from multiple projects, or from only one project, or from a particular component of a project.

Before you begin

You must be a **Jira administrator** or a **board administrator** for the board to modify its configuration.

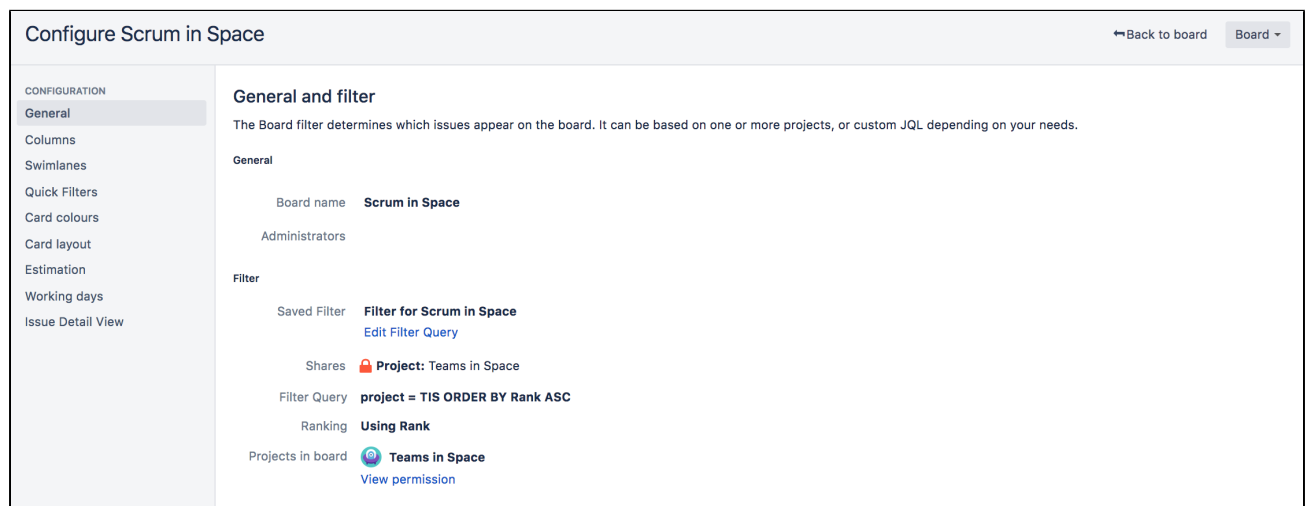## Choosing a different filter for your board

1. Go to the desired board and select **Board** > **Configure**.
2. Click the name of the filter displayed in the **Saved Filter** field. A pencil icon ✎ will appear when you hover over it.
3. Choose a different filter for your board. Press the **Enter** key when finished.

Screenshot: the 'Board Configuration' screen — 'General' tab
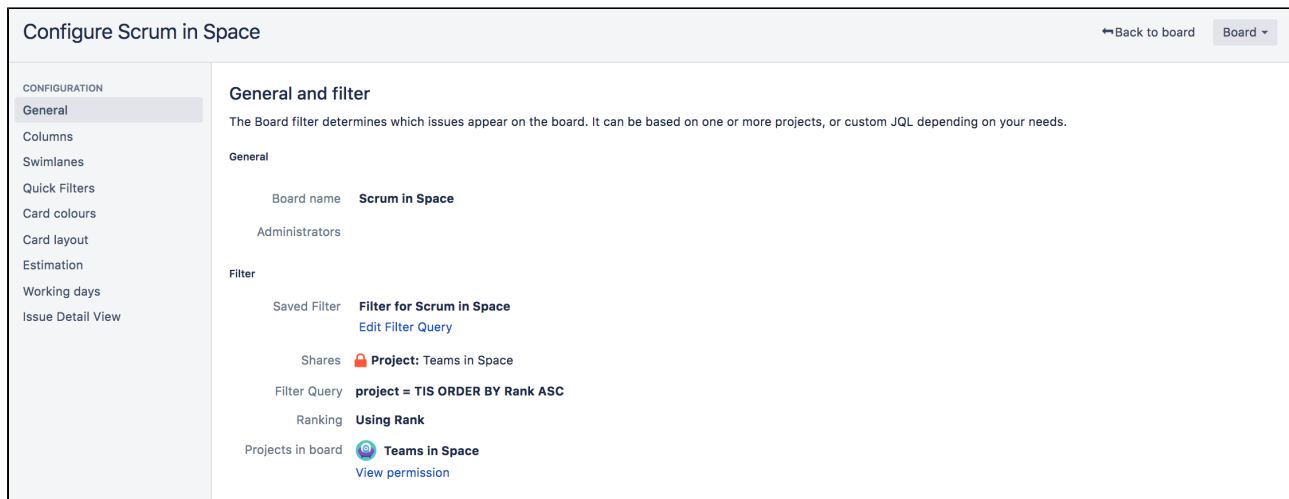
Editing a board's filter

1. Go to the desired board and select **Board** > **Configure**.
2. In the **General** tab:
   - To change the filter's JQL query, click **Edit Filter Query**. For more details, see the Jira documentation on JQL.
   - To change the filter's name, description, or shares, click **Edit Filter Shares**. For more details, see the Jira documentation on issue filters.

(Kanban only) Adding a sub-filter

If you are using a Kanban board, you can add a sub-filter that refines the issues returned by your board's filter. The sub-filter enables the **Kanban board** to show different data to the **Reports**.

The default sub-filter is:

```
fixVersion in unreleasedVersions() OR fixVersion is EMPTY
```

... this will result in issues released using the 'Release' button no longer appearing in the **Active sprints** (although the reports in **Reports** will continue to show them).

To add a sub-filter to your board:

1. Go to the desired board and select **Board** > **Configure**.
2. Edit the **Kanban board sub-filter** field and enter a JQL query. For information on JQL syntax, see JQL (Jira Admin documentation).

(Kanban only) Hiding completed issues

To improve your board's performance, all issues in the **Done** column which weren't updated within the last 2 weeks are hidden. You can change this time period, or choose to always show all issues.

To hide completed issues:

1. Go to the desired board and select **Board** > **Configure**.
2. In the **Hide completed issues** field, select the retention period.
   Issues that weren't updated within this period will be hidden.

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Enabling ranking

Enabling ranking allows you to rank issues on a board by drag-and-drop, and to create sprints on Scrum boards. It also enables Jira Software to group sub-tasks together underneath their parent issue.

Before you begin

You must be a **Jira administrator** or a **board administrator** for the board to enable ranking.

Enabling ranking

1. Go to the desired board and select **Board** > **Configure**.
   Note that only the administrator of a board (or a person with the 'Jira Administrators' global permission) can configure a board.
2. Click the **Add Rank** button. This will append the following to your filter's query:

```
ORDER BY Rank ASC
```

Notes:

- Jira Software automatically creates a custom field called **Rank**, of type **Global Rank**. Please ensure that your Jira instance contains only one custom field called **Rank**, and that the **Rank** field is assigned to the global context (i.e. not to specific issues or projects).
- The sort order must be ascending. Descending sort order is not supported.

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Configuring columns

The vertical columns in both *the Active sprints of a Scrum board* and *the Kanban board* represent the workflow of your board's project.

The default columns in the Active Sprints of a Scrum board are **To Do**, **In Progress**, and **Done**.

The default columns on a Kanban board are **Backlog**, **Selected for Development**, **In Progress**, and **Done**.

You can add, delete, rename, or move these columns if you wish. You can also choose which Jira workflow status or statuses each column is mapped to, and whether any constraints apply to each column.

On this page:

- Before you begin
- Editing columns
- Mapping statuses to columns
- Editing the mapping of Jira workflow statuses to columns of a board
- Adding a new status
- Deleting a status
- Setting column constraints
- Enabling the Kanban backlog
- Next steps

Before you begin

You must be a **Jira administrator** or a **board administrator** for the board to configure its columns.

Editing columns

1. Go to the desired board and select **Board** > **Configure**.
2. Click the **Columns** tab.
3. Edit the columns as described in the following table and the screenshot (below).

Add a new column	Click the **Add Column** button at the right of the page. A new column named '**New Column**' is added in the second to the last column position. If you are using Simplified Workflow, then a new status will automatically be created to match your new column.
Change the name of a column	Click in the name area of the column, modify the existing name, and press **Enter**.

Show the 'Days in column' indicator on cards	Click the **Days in column** checkbox. This shows a series of dots on each card (up to the width of the card or a maximum of 32), representing the number of days that the issue has been in the column. This can help you see issues that are stagnating — this is particularly useful when your board is displayed as a wallboard. Note that if you move an issue back to one of the columns that you moved it into previously, the indicator reflects a cumulative value of the number of days the issue stayed in that column. For example, you move an issue to the 'In Progress' column and it stays there for 2 days. You then move the issue to the 'Code Review' column and it stays there for 1 day. During code review, you receive feedback, which requires more development work for the issue, so you move the issue back to the 'In Progress' column and it stays there for 1 day. In this example, the indicator reflects the cumulative value of 3 days for the issue in the 'In Progress' column. Disabling this indicator can improve performance. If you have a large instance (i.e. 300,000+ issues, 100+ projects, 100+ boards, or 100+ open sprints), we recommend that you disable this. This indicator is disabled by default for Scrum boards and enabled by default for Kanban boards.
Delete a column	Click **delete** in the "Drag to rearrange or delete" text at the top of the column. Any Jira [workflow](#) statuses that had been mapped to the deleted column are moved back to the '**Unmapped**' column.
Move a column	Hover over the top of the name of a column, then drag the column left or right to its new position. Wait until other columns have shifted position before dropping the selected column to its new position.

Screenshot: the 'Board Configuration' screen — 'Columns' tab.

Each column has a blue, yellow, or green color, as shown in the screenshot above.

- The single left-most column will always be blue, representing items in 'new' state.
- The single right-most column will always be green, representing items in a 'successful' state.
- The middle column (or columns if you add more) are always yellow, representing items in progress.

You will see these colors shown in a number of places in Jira Software, e.g. in [gadgets for JIRA applications](#).

Mapping statuses to columns

By default, a board's columns are mapped to the default Jira statuses, as shown in the following tables. You can change this if you wish. For example, if you are using additional, customized JIRA statuses, you will probably want to map them to appropriate columns in your board.

If your board's project is using the Jira default workflow:

Default column	Default mapped Jira statuses
To Do	Open, Reopened
In Progress	In Progress
Done	Resolved, Closed

If your board's project is using Simplified Workflow:

Default column	Default mapped Jira statuses
To Do	To Do
In Progress	In Progress
Done	Done

Editing the mapping of Jira workflow statuses to columns of a board

1. Go to the desired board and select **Board** > **Configure**.
2. Click the **Columns** tab.
3. Change the mapping statuses, as described in the following table and screenshot (above).

Map a status to a column	Drag a status from the **Unmapped** column to the appropriate column on the right.
Unmap a status from a column	Drag a status from its current column on the right to the **Unmapped** column on the left.
Change the column mappings of a status	Drag a status from its original column to its relevant new column.

Note, all statuses configured in the Jira server are available from the board Configuration page. However, some statuses (in particular, custom statuses) may not be available for issues on your board if the Jira workflows used by these issues do not utilize those statuses.

Adding a new status

If your board is using Simplified Workflow:

1. Go to the desired board and select **Board** > **Configure**.
2. Click the **Columns** tab.
3. Click the **Add Status** button at the right of the page.
   ⓘ Note that the **Add Status** button is only available if you have the Jira '**Project Administrator**' permission for this board's project. See Permissions overview for more information.

Deleting a status

You can only delete a status if:

- you are the 'Project Administrator' for this board's project(s), and
- there are no issues that currently have that status.

If any other workflows are using this status, then the status will be removed from your workflow, but not deleted.

If your board is using Simplified Workflow:

1. Go to the desired board and select **Board** > **Configure**.
2. Click the **Columns** tab.

If your board is *not* using Simplified Workflow, you can only delete a status via the Jira administration interface. See Workflows for more information.

3. Drag the status to the **Unmapped Statuses** c
    olumn.
4. Click the **Remove Status icon (x)** for the
    status you wish to remove.

Setting column constraints

Setting constraints on each workflow state is a crucial part of Kanban, so that you can ensure that work is continually flowing through the pipeline. You may also find it handy in Scrum, so that you can see bottlenecks early, and adjust the scope of your sprint.

Constraints trigger a visual indicator when a maximum or minimum number of issues appear in a column. You can specify constraints for all columns, or just some of them. If a column constraint is exceeded, then the column will change color, as follows:

- Column colored red — Maximum number of issues exceeded
- Column colored yellow — Minimum number of issues not met

Please note:

- The values of the column constraints will appear at the top of each column you set them for. These values don't have any impact on the number of issues displayed in the columns.
- Column constraints apply to the total number of issues in a column, regardless of whether issues are hidden by a Quick Filter.

To set column constraints of a board:

1. Go to the desired board and select **Board** > **Configure**.
2. Click the **Columns** tab.
3. Edit the constraints, as described in the following table.

Enable column constraints	In the **Column Constraint** drop-down, select either **Issue Count** or **Issue Count, excluding sub-tasks**. Note, enabling column constraints will show the issue count in each column. You may want to exclude sub-tasks (particularly Scrum teams), so you can constrain the number of stories for your team, rather than the number of individual tasks.
Set a column's 'Min' or 'Max' constraint	Edit the fields at the top of the desired column: - Enter a minimum value in the yellow box. - Enter a maximum value in the red box. Delete an existing value to remove the constraint.

Note, you can also remove constraints for all columns. In the **Column Constraint** drop-down, select **N one** .

Enabling the Kanban backlog

ⓘ You must be a **Jira administrator** or a **board administrator** for the board to enable the Kanban backlog.

Managing your backlog in the first column of your Kanban board is easy to do — as long as there are only a few issues in your backlog. As your backlog grows, viewing and scrolling through these issues can become difficult.

By using the Kanban backlog, you get a bigger backlog that gives you an optimized list view of the issues you're creating and ranking for your team. With versions and epics displayed in separate panels, you can also easily assign issues to specific releases, and filter these issues by epics that they belong to. The backlog is perfect for planning story development, without distracting your team from their work-in-progress in the Kanban board.

To enable the Kanban backlog:

1. Go to the desired board and select **Board** > **Configure**.
2. Click the **Columns** tab. The Column configuration page displays, with the *Kanban Backlog* as the leftmost column.
   Screenshot: the 'Board configuration' screen — 'Columns' tab, columns section
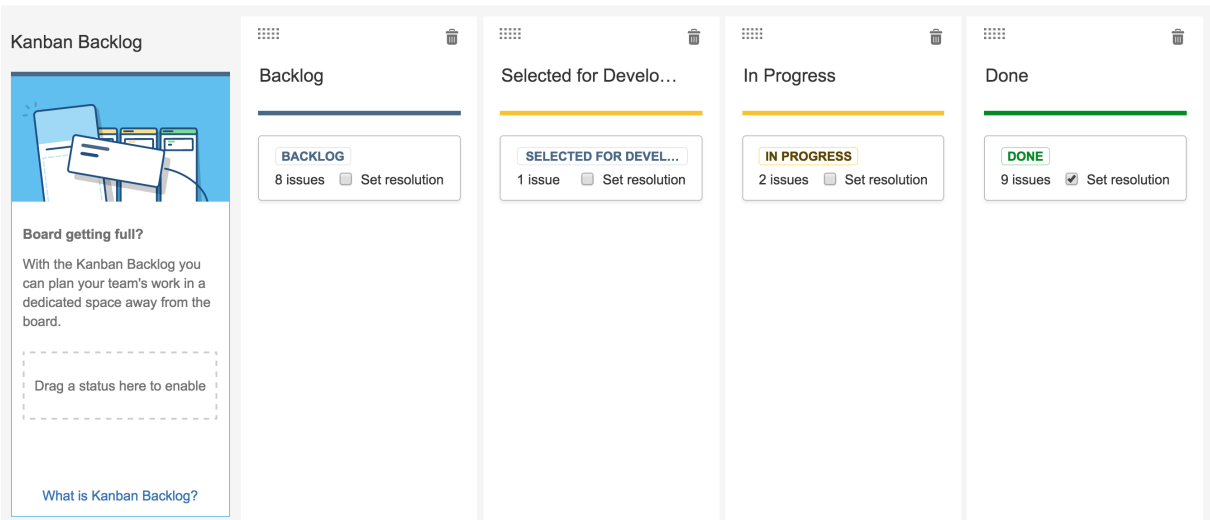
3. In the columns section, drag a status to the Kanban Backlog column. Typically, this would be the column for which you plan work for your team. Once you do this, the backlog will be enabled.

To change how epics are displayed in Jira:

After enabling the Kanban backlog, all epics are moved to the Epics panel in the backlog, and not displayed as cards on the board like all other issues. If you don't like it, you can use the 'Epics panel' toggle above the columns to disable the epics panel.

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

- Using Plan mode
- Using your Kanban backlog
- Using the Simplified Workflow
- Transitioning an issue

Using the Simplified Workflow

Every board can represent one or more projects; however, the Simplified Workflow can only be used if a board represents a single project. Every project uses either a **Jira Workflow** or the **Simplified Workflow** to control the transitioning of issues from one status to another. The workflow determines which statuses are available.

Your board will be using the Simplified Workflow if your board was created as part of creating a project, (including via the Jira Software Getting Started page).

Before you begin

You must be a **Jira administrator** or a **board administrator** for the board to switch it to the Simplified Workflow.

Benefits of using the Simplified Workflow

The Simplified Workflow offers the following benefits over a JIRA workflow:

Simplified Workflow	JIRA Workflow
- Has three default steps for Scrum boards: **To Do, In Progress, Done.** - Has four default steps for Kanban boards: **Backlog, Selected for Development, In Progress, Done.**	- Has more steps than are typically needed on a board (e.g. see the default Jira workflow).
- Allows issues to be dragged freely between columns.	- Has workflow "conditions" that prevent issues from being dragged freely between all columns.
- Displays no screens on any transitions – all transitions will happen instantly. - Automatically sets a resolution of '**Done**' when issues are transitioned to the 'successful' column, i.e. statuses mapped to the successful column (see Configuring columns).	- Displays screens for **Resolve Issue**, **Close Issue**, and **Reopen Issue**.
- Can be edited via the board configuration (see *Adding a new status* on the Configuring columns page), provided that you are the project administrator for the project that is on the board.	- Can only be edited via workflow configuration, see Workflows for more information.

For information about workflow steps , transitions, and conditions, see Workflows for more information.

Switching to Simplified Workflow

> ⓘ **How can I tell if my board is using Simplified Workflow?**
>
> To check this, view the **Columns** configuration for your board (described below) — If you are using the Simplified Workflow, you will see the words 'Using Simplified Workflow' (see screenshot). Otherwise, you will see the 'Simplify workflow' button or the words 'Simplified Workflow unavailable'.

If your project is currently using a Jira workflow, you may want to switch to the Simplified Workflow. You will then be able to easily add new statuses from within Jira Software . Note, the default steps for the Simplified Workflow (described in table above) *will not* be created when you switch from a Jira workflow. Your existing workflow steps will be included in the workflow instead. The default steps are only created if the Simplified Workflow is chosen when the board and project are created in Jira Software.

Project pre-requisites

You will only be able to switch to Simplified Workflow if:

- There is only one project being viewed by your board (to check this, look at the board's filter)
- That project uses a Jira workflow scheme, which only has one workflow for all issue types
- Your workflow only uses Post Functions, Validators, and Conditions, which are provided by Atlassian (not any of which are provided by apps)
- The existing workflow has at least one outgoing transition for each status

To switch to Simplified Workflow:

1. Before you begin, back up your Jira data, see Backing up data (Jira Admin documentation). Switching to the Simplified Workflow will migrate all of the project's issues to the new workflow, which may take some time. The project's original workflow and scheme not be changed, so any other projects using these will not be affected.
2. Go to the desired board and select **Board** > **Configure**.
3. Click the **Columns** tab.
4. Click the **Simplify workflow** button, as shown in first screenshot below. If this button does not display, check that your project meets the pre-requisites described above.
5. Your project(s) will switch to use Simplified Workflow, as shown in the second screenshot below. The **Add Status** button will now be shown.

Screenshot: the 'Board Configuration' screen ('Columns' tab) — this board is not using Simplified Workflow:
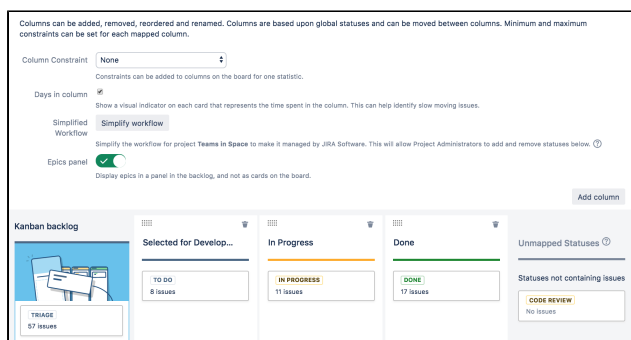
Screenshot: the 'Board Configuration' screen ('Columns' tab) — this board is using Simplified Workflow:
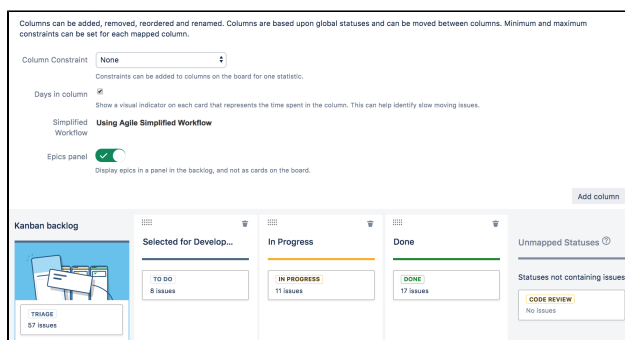
Next steps

ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

- Read Configuring columns to learn how to configure other settings for your columns, e.g. mapping statuses to columns.
- Read Transitioning an issue for information on how issues transition from column to column.
- Read Configuring filters for information on how to check a board's filter.

Configuring swimlanes

A swimlane is a horizontal categorization of issues in the Active sprints of a Scrum board, or on a Kanban board. You could use swimlanes to help you distinguish tasks from different workstreams, users, application areas, etc.

Before you begin

You must be a **Jira administrator** or a **board administrator** for the board to configure its swimlanes.

Choosing a different type of swimlane

You can choose to set up your swimlanes in a variety of ways, as shown in the following table.

Base your swimlanes on...	Explanation
Queries	One JQL query per swimlane (see below for examples). By default, two swimlanes will be created: - **Expedite** — this swimlane is based on the following JQL query: `priority = Blocker` - **Everything Else** — this swimlane is always at the bottom of the screen, and cannot be deleted. It acts as a "catch-all" for issues that do not match the JQL of any of the swimlanes above it, hence it has no JQL specified. You may want to create additional swimlanes that map to other values of your Jira 'Priority' field, or use a different field to categorize your swimlanes.
Stories	One parent issue per swimlane (i.e. each swimlane contains all of the parent's sub-tasks), with issues that have no sub-tasks appearing below.
Assignees	One assignee per swimlane, with unassigned issues appearing either above or below the swimlanes (your choice).
Epics	One epic per swimlane, with issues that don't belong any to epics appearing below the swimlanes. If you want to change the order of the swimlanes on your board, navigate to the **Backlog** of the board and drag and drop the epics as desired.
No Swimlanes	No horizontal categorization of issues in the Active sprints of a Scrum board, or on a Kanban board.

To choose a different type of swimlane:

1. Go to the desired board and select **Board** > **Configure**.
2. Click the **Swimlanes** tab.
3. In the **Base Swimlanes on** drop-down, select either **Queries**, **Stories**, **Assignees**, or **No Swimlanes**, as described above.

Screenshot: the 'Board Configuration' screen — 'Swimlanes' tab

Swimlanes

A swimlane is a row on the board that can be used to group issues. Swimlane type can be changed below and will be saved automatically.
Note: queries will not be lost when changing to another swimlane type.

Base Swimlanes on [Queries ⇕]

Group issues by custom queries. Queries are based on JQL that is additional to the saved filter of this board.

Name	JQL	Description	
	✅		Add
TIS Developer Love	component = Add-ons		Delete
Expedite	priority = Blocker	All issues with the highest priority.	Delete
Everything Else			

Modifying your Query-based swimlanes

If your swimlanes are based on JQL queries (rather than on stories or assignees), you can create, delete, or change them:

1. Go to the desired board and select **Board** > **Configure**.
2. Click the **Swimlanes** tab.
3. If your swimlanes are based on **Queries**, you can edit your swimlanes, as described below and the screenshot above.

Add a new swimlane
In the blue area, type the **Name**, **JQL**, and optional **Description**, then click the **Add** button. Your new swimlane is added in the top swimlane position.

Change the name of a swimlane
Click in the **Name** area of the swimlane, modify the existing name, and click the **Update** button.

Change the JQL of a swimlane
Click in the **JQL** area of the swimlane, modify the existing JQL, and click the **Update** button.

See the examples below for some suggestions. For information on JQL syntax, see JQL (Jira Admin documentation).

Note, the JQL 'ORDER BY' clause is not used by the swimlane, as it defaults to order by rank.

Delete a swimlane
Click the **Delete** button at the right of the swimlane.

Move a swimlane
Hover over the vertical 'grid' icon, then drag and drop the swimlane up or down to its new position.

Some example JQL queries for your swimlanes:

- Show all issues that belong to a particular component, e.g. 'User Interface':

```
project = "Angry Nerds" AND component = "User Interface"
```

- Show all issues that are due in the next 24 hours:

```
due <= "24h"
```

- Show all issues that have a particular priority, e.g.:

```
priority = "Minor"
```

and

```
priority = "Major"
```

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Configuring Quick Filters

Quick Filters allow you (or anyone else using this board) to further filter the collection of issues appearing on a Scrum board or Kanban board. Use Quick Filters to switch between different issue types (e.g. show only bugs), or to show team-specific views of a common backlog.

Before you begin

You must be a **Jira administrator** or a **board administrator** for the board to configure its Quick Filters.

About the default Quick Filters

By default, a board contains two Quick Filters, called 'Only My Issues' and 'Recently Updated':

Default Quick Filter	Default JQL	Notes
Only My Issues	`assignee = currentUser()`	Displays issues assigned to the person who is currently viewing this board
Recently Updated	`updatedDate >= -1d`	Displays issues that have been updated in the last 24 hours

Editing Quick Filters

You can create additional Quick Filters or edit existing Quick Filters. Any additional Quick Filters that you create will appear as buttons next to the 'Only My Issues' and 'Recently Updated' buttons on the board.

1. Go to the desired board and select **Board** > **Configure**.
2. Click the **Quick Filters** tab.
3. Edit the Quick Filters, as described below and in the screenshot (below).

   Add a new Quick Filter
   In the blue area, type the **Name**, **JQL**, and a **Description** (optional), then click the **Add** button. Your new Quick Filter will be added in the top Quick Filter position.

   Change the name of a Quick Filter
   Click in the **Name** area of the Quick Filter, modify the existing name, and click the **Update** button.

   Change the JQL of a Quick Filter
   Click in the **JQL** area of the Quick Filter, modify the existing JQL, and click the **Update** button. See the examples below for some suggestions. For information on JQL syntax, see JQL.

   Delete a Quick Filter
   Click the **Delete** button at the right of the Quick Filter.

   Move a Quick Filter

   Hover over the vertical 'grid' ⁞ icon, then drag and drop the Quick Filter up or down to its new position.

   When this board is used, the top-most Quick Filter appears in the left-most position.

Screenshot: the 'Board Configuration' screen — 'Quick Filters' tab

Quick Filters can be used to further filter the issues in the board based on the additional JQL query.

Name	JQL	Description	
	✅		Add
Only My Issues	assignee = currentUser()	Displays issues which are currently assigned to the current user	Delete
Recently Updated	updatedDate >= -1d	Displays issues which have been updated in the last day	Delete

Sample JQL you might wish to use for your Quick Filters:

- Show all issues that are assigned to members of the "bugfix" group:

```
assignee in membersOf("bugfix")
```

- Show all issues of type 'Bug':

```
type = "Bug"
```

- Use an issue filter in your Quick Filter:

```
savedfilter = "My Filter"
```

Next steps

> (i) **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

- Read Configuring a board to learn how to configure other parts of your board, like columns or swimlanes.
- Read JQL for information on JQL syntax.

Customizing cards

Don't rely on people clicking cards to view the issue details. Customizing the cards for your boards can help you bring the right information to your team's attention, at a glance. You can **change the card colors** to help people quickly identify cards on your board as being of a particular issue type, priority, assignee, or — thanks to the power of JQL — practically anything you choose. You can also **add up to three additional fields** to display on cards in the Backlog and Active sprints of your Scrum board.

Before you begin

You must be a **Jira administrator** or a **board administrator** for the board to customize its cards.

Configuring card colors

You can base your card colors on issue types, priorities, assignees, or JQL queries. Once you have chosen a method, you can change or delete the colors for each type of card. This can be configured per board (not globally).

1. Go to the desired board and select **Board** > **Configure**.
2. Click **Card Colors** and change the **Colors based on** drop-down as desired. If you change to a different method of card coloring, your settings for the old method will be retained so you can switch back to them later if you wish.

Base your card colors on...	Explanation
Issue types	One color per issue type. A default color will be allocated to every issue type that matches issues on the board. Note, the issue type must already exist in your project to configure the color for it. Also, the color values won't display until an issue is created on your board. You need to create an issue on your board first, to configure the colors for the issue types. Default issue types and colors: Improvement, Task, New Feature, Bug
Priorities	One color per priority. The default colors are the same as used for priorities in Jira.
Assignees	One color per assignee. A default color will be allocated to every user who is or has been an assignee of issues on this board. Note, the color values won't display until an issue is created on your board. You need to create an issue on your board first, to configure the colors for the assignees.

Queries	One color per JQL query. You can specify whatever queries you wish. Issues that do not match any of your JQL queries will be shown in grey. See JQL for more information about JQL syntax.
	Example queries:
	- All issues that belong to a particular component, e.g. 'User Interface':
	`project = "Angry Nerds" AND component = "User Interface"`
	- All issues that are due in the next 24 hours:
	`due <= "24h"`
	- All issues that are blocked by other issues, or that have no blockers:
	`issueLinkType = "is blocked by"`
	`issueLinkType != "is blocked by"`

3. Once you have picked a method to base your colors on, customize the colors as follows:

- **Pick a different color for a card** — Click the **Color** square, e.g.
- **Delete a card color** — Click **Delete**. This effectively resets the card to the default color. If you still have issues matching that card type, refresh the screen to reset the deleted card color to the default color).
- **Move a card color (Query-based only)** — Hover over the vertical 'grid' icon, then drag and drop the color up or down to its new position.

   Note: For Query-based colors, the order is important, as each issue will be colored according to the first query that it matches (e.g. if your first row first row has query "issuetype = bug" and is colored red, and your second row has query "assignee = kevin" and is colored green, then bugs assigned to Dave will appear red).

Adding fields to cards

You can configure cards on a board to display up to three additional fields. The fields can be different for the Backlog and Active sprints, if you are using a Scrum board.

1. Go to the desired board and select **Board** > **Configure**.
2. Click **Card layout**, then add or delete fields as desired.

Screenshot: Cards displaying additional fields in the Backlog
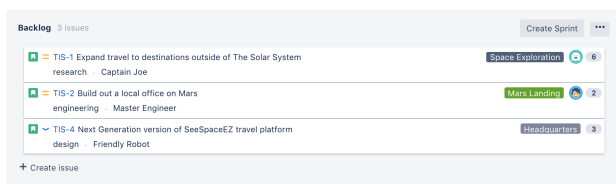
Screenshot: Cards displaying additional fields in Active sprints
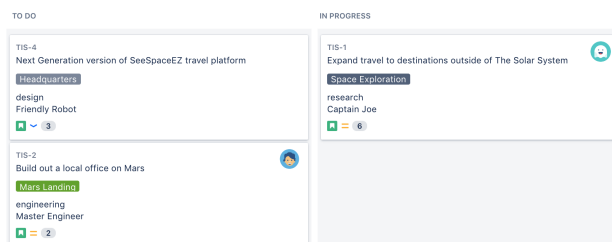
Adding Days in column to your cards

This shows a series of dots on each card (up to the width of the card or a maximum of 32), representing the number of days that the issue has been in the column. This can help you see issues that are stagnating — this is particularly useful when your board is displayed as a wallboard.

KANBA-4

Work items are ranked in priority order (from top to bottom) >> Try dragging this card over the card below to rank its

KANBA-5

Work items flow through different stages from left to right >> Try dragging this card to "Selected for Development"

ⓘ If you move an issue back to one of the columns that you moved it into previously, the indicator reflects a cumulative value of the number of days the issue stayed in that column.

To add **Days in column**:

1. Go to the desired board, then click **Board** > **Configure**.
2. Click the **Columns** tab.
3. Select **Days in columns.**

   ⓘ If you have a large instance (i.e. 300,000+ issues, 100+ projects, 100+ boards, or 100+ open sprints), we recommend that you don't enable this option as it might affect performance. The option is disabled by default for Scrum boards and enabled by default for Kanban boards.

Dots on cards	Number of days in column
•	1 day
••	2 days

●●●	20 days
●●●●	25 or more days

Printing issue cards

Whether you're planning work or working on issues for Scrum or Kanban projects, it may also be good to print out these issue cards. You and your team can use the printed cards on a physical board, which can replicate your board on Jira Software. If you want to, you can print a single issue card or multiple issue cards. You can also print all issue cards in your current board.

The printed issue cards include the following issue details:

- Summary
- Issue type
- Issue key
- Issue priority
- Estimate
- Assignee
- Epic (optional in Active sprints/Kanban board)
- Version (when printing from the backlog)
- Up to 3 extra fields, depending on your card layout configuration

The printed issue cards fit on A4-, A3-, or Letter-sized pages in both portrait and landscape modes.

See Printing issue cards for more information.

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

- Read Printing issue cards to learn how to print issue cards and use a physical replication of your board in Jira Software.

Configuring estimation and tracking

The estimation statistic (e.g. story points, business value, etc) and the time tracking settings (remaining time estimates) can be customized to suit how you estimate and track work in your project.Many Scrum teams separate *estimation* (which is used for measuring the size of a backlog and calculating velocity) from *tracking* (which is often the burndown of hours used during the Sprint to make sure that they are on track to complete the stories during the Sprint period), and use different units for each. A common approach is to *estimate* tasks in Story Points, then *track* tasks using hours. Jira Software gives you the flexibility to set your estimation and tracking statistics differently, depending on what best suits your team.

Before you begin

- You can only configure estimation and time tracking for Scrum boards.
- You must be a **Jira administrator** or a **board administrator** for the board to configure estimation or time tracking for it.

Setting estimation statistic and time tracking

1. Go to the desired board and select **Board** > **Configure**.
   Note, only the administrator of a board (or a person with the 'JIRA Administrators' global permission) can configure a board.
2. Select the **Estimation** tab. The Estimation page will be displayed.

Estimation

Issues can be estimated when in the Backlog to get an idea of how much work is being committed to in a sprint. Read more about estimation and tracking.

Estimation Statistic   Story Points

Estimate issues in the Backlog by entering values for **Story Points**. Your velocity from sprint to sprint will be measured against these estimates.

Time Tracking   ● None

Issues will burn down their **Story Points** value upon completion.

○ Remaining Estimate and Time Spent

Track time against issues using JIRA's **Remaining Estimate** and **Time Spent** fields. The Burndown Chart will be based on these two values.

3. In the **Estimation Statistic** field, select one of the following options:

Estimation statistic	Explanation
Story Points	Estimation will be based on the number of Story Points per issue. This is the most commonly used option. By default, the Story Points field is only available to issues of type 'Story' or 'Epic' (not 'Bugs' etc). If you want to change this, do the following: 1. Associate the Story Points field with other issue types, see custom field context (Jira Admin documentation). 2. Specify the screens that the Story Points field should be displayed on, see Defining a screen (Jira Admin documentation).
Original Time Estimate	Estimation will be based on the Jira '**Original Time Estimate**' field (see Logging work on issues for more information). By default, this is specified in minutes, but you can use hours, days, or weeks, depending on your Jira system configuration. See Configuring time tracking (Jira Admin documentation).
Business Value	Estimation will be based on the Business Value of each issue.
Issue Count	Estimation will be based on the number of issues in the sprint. The **'Estimate'** field will not be editable.

<Custom Field>	Estimation can be based on any numeric custom field in your Jira system.

4. In the **Time Tracking** field, select one of the following options:

Tracking Statistic	Explanation
None	Tracking will be based on the Estimation Statistic.
Remaining Estimate and Time Spent	Tracking will be based on the Jira '**Remaining Estimate**' and **'Time Spent'** fields (see Logging work on issues for more information). By default, these fields are specified in minutes, but you can use hours, days, or weeks, depending on your JIRA system configuration, see Configuring time tracking (Jira Admin documentation). Note that this is fundamentally different from using the Estimation Statistic for burndown, in that values do not burn down when an issue is completed — instead, values only burn down when users enter **Time Spent** or set the **Remaining Estimate** to a new value.

More information about estimation and time tracking

Product teams often need to be able to estimate how long a product will take to deliver. This is difficult because the backlog may stretch many months into the future, so the team can only provide a very rough estimate in conditions of uncertainty without wasting days breaking the work down. However, from sprint to sprint, as they work through the stories, the team will develop a cadence of completing <x> units of work they had 'roughly estimated', i.e. their *velocity* .

This means that they can accurately estimate how long portions of the backlog will take to get done with simple rough estimates. However, to make this work, the team needs to estimate stories with a consistent level of uncertainty. The team also needs to track the amount of estimation units they have actually fully completed from sprint to sprint, because this number tells us with relative certainty how much we can fit into each future sprint.

How the Estimation Statistic and Tracking Statistic affects your project

In Jira Software, you can choose which type of units (e.g. Story Points, Issue Count) will be used for estimating and tracking issues. You do this by choosing an Estimation Statistic, then choosing to either use the same units for your Tracking Statistic or to use time-tracking. Each board can have a different type of Estimation Statistic and Tracking Statistic.

- The type of Estimation Statistic you select affects the units that are used by the **Estimate** field, which appears at the right of each issue in the Backlog:

- The type of Tracking Statistic you select affects the units that are used by the **Remaining** field, which appears at the bottom of each issue in Active sprints:
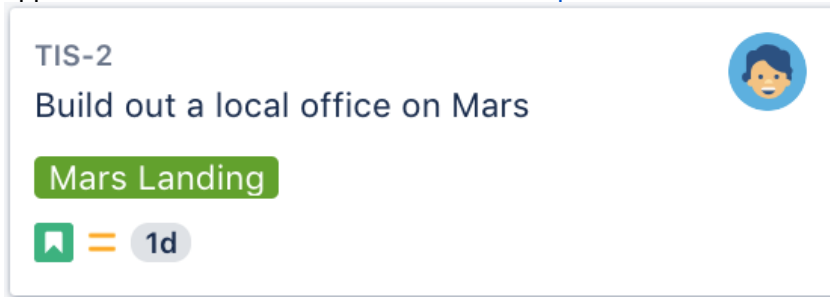
How you can view your Velocity and Burndown

A team's velocity is based on the Estimation Statistic — i.e. for each sprint, the velocity is the sum of the Estimation Statistic for completed stories. Velocity is shown in the Velocity Chart and also on the Sprint Report, in the Estimate Statistic column header of the "Completed Issues" table (e.g. "Story Points (12)" means that 12 Story Points were completed in that sprint). Please note that the values for each issue are recorded at the time when the issue moves into the sprint. Changing the Estimate value afterwards will not be reflected in the Sprint Report, but will be shown as scope change in the burndown. Velocity is also used in the Version Report, to predict Release Dates.

The Burndown Chart is based on the Tracking Statistic. If you are using Story Points as your Tracking Statistic, then the Burndown Chart shows the Story Points per story (i.e. stories burning down the Estimate Statistic are only burnt down on the graph as they are completed); whereas if you choose the Time-tracking option, you are shown partial burndown (i.e. the number of hours currently used and remaining each day).

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

- Learn more about estimating issues: Estimating an issue
- Learn about some of the charts that estimation and time tracking affects: Burndown Chart, Velocity Chart, Version Report

Configuring the issue view
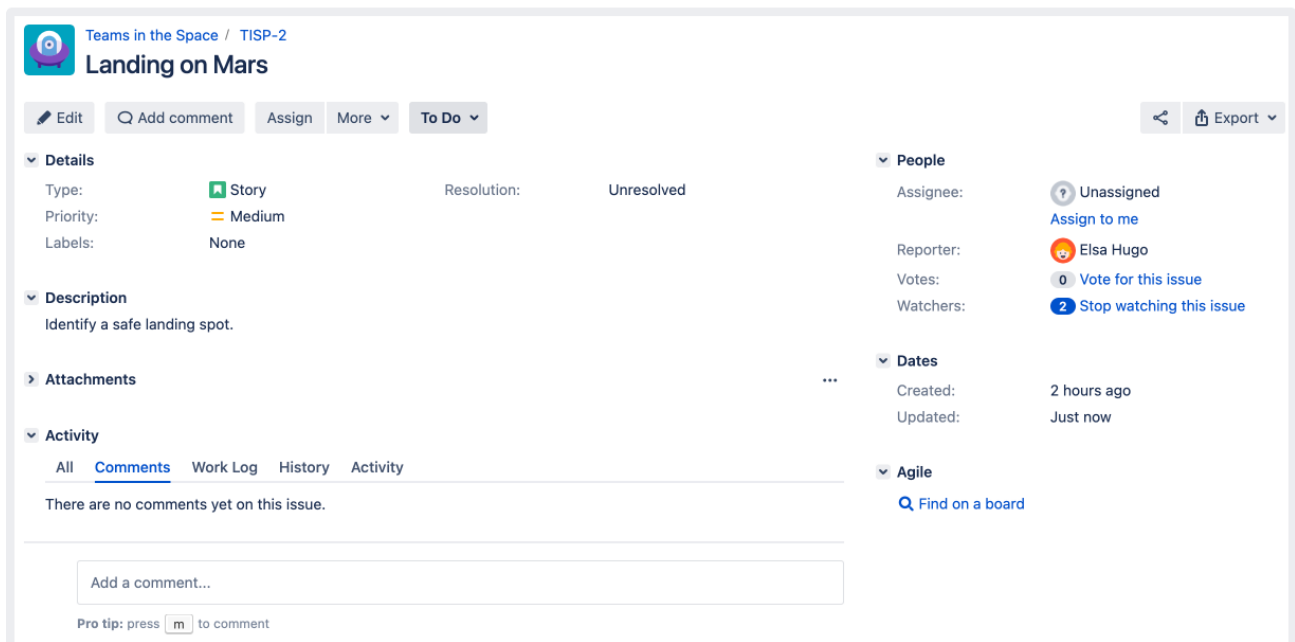
Jira issue view can be configured to display various system and custom fields that describe the issue. You can customize this view by adding new fields, deleting fields, or changing their positions. For example, you may want to show an issue's Resolution, Priority, Labels, Security Level, etc.

On this page:

- Board configuration
- Project configuration
  - Making the added field editable in the issue view
- Next steps
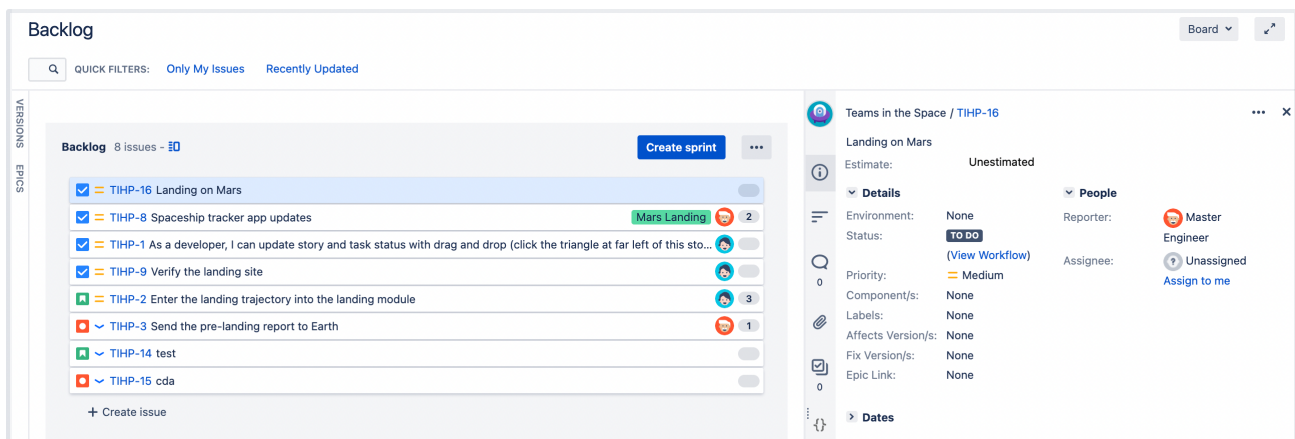
> ℹ️ For all of the following procedures, you must be logged in as a user with the **Jira admin** or a **board admin** permissions. For details, see Permissions overview.

Here's what the full issue view looks like (if you open the issue by selecting its key; in this example, TIHP-16).

And here's what the issue details view looks like (if you open the issue from the backlog or board):

You can configure the fields in the issue details view from two places:

- Board configuration

- Project configuration

Let's see what you can do there and how your changes will impact the layout of the issue view.

Board configuration

Use the board configuration to:

- add more fields to the issue details view
- change the positions of fields
- hide some fields from the issue details view
- add fields to issues on all boards in your project. To view the fields when opening issues on the boards, you'll need to add the fields through the board configurations for each board.

> (i) Note that the fields you add to the issue details view through the board configurations won't be editable. The following sections include instructions on how to play with the fields in this list, as well as how to make them editable.

Adding, moving, or deleting a field

1. Go to the desired board and select **Board** > **Configure**.
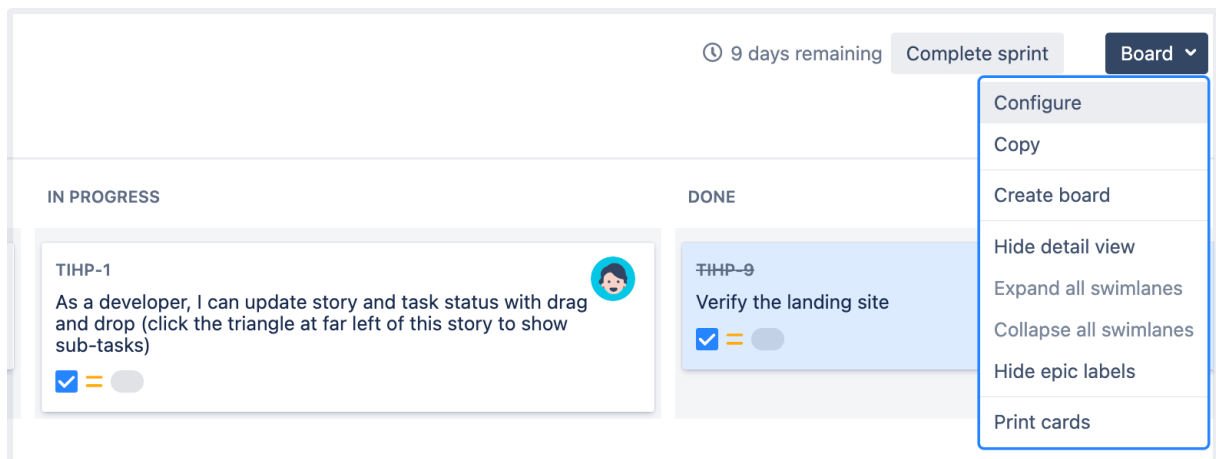
2. In the left-side menu, select **Issue Detail View**. You'll see different groups of fields: General fields, Date fields, People, and Links. The fields that you can see in these groups make up the issue view.
3. To add a new field, select the field from one of the dropdown menus, and then select **Add.**
4. If you want to hide a field from the issue details view, select **Delete**.

5. To change the position of the field in the issue details view, use the vertical grid icon to drag and drop the field up or down in the list.

Add, delete or reorder fields on the Issue Detail View.

General fields

Field name

Bug Count	⇕	Add
Status		Delete
Priority		Delete
Component/s		Delete
Labels		Delete
Affects Version/s		Delete
Fix Version/s		Delete
Epic		Delete

Date fields

Field name

Borrowed on	⇕	Add
Created		Delete
Updated		Delete

ⓘ The **Bug Count** field will be added to the issue details view, but the field won't be editable until you add it to the appropriate issue screen.

You'll see the added field only if you open the issue on the board or from the backlog. To view the field in the full issue view, you need to associate that field with the relevant issue type and make sure that the field isn't hidden. For details, check the following articles:

- Specifying field behavior for built-in fields
- Adding custom fields for custom fields

There's one important thing you should know about the **Story points** field and how it's different from the **Original story points** field.

You can add the **Story points** field only through the Screens configuration. You'll see this field with its original name **Story points** in the full issue view. Also, if you open the issue on the board, you'll see the **Estimate** field that is the same as **Story points**. The behavior of the **Estimate** field depends on the estimation configuration on the board:

- If the estimation configuration is set to story points, the **Estimate** field in the issue view will contain a numeric value indicating the number of set story points.
- If the estimation configuration is set to original time estimate or original story points, the **Estimate** field will contain the value **Unestimated**.

- If the estimation configuration is set to issue count, you won't see the **Estimate** field in the issue view on the board.

The **Original story points** field can be added both through the board configuration and the Screens configuration. So, you can manage this field in the same way as the **Sprint** field in the above example.

Why is it so and what's the difference between these concepts? This is the Jira way: you can add the **Story points** field to the issue view only by adding it to the issue screen schemes.

The difference between story points and original story points is the following: if you're using story points as the estimation unit in your project, you can use original story points as initial, rough estimates of the work you're planning to do.

Then, you can use original story points as references when you start breaking down and providing more detailed estimates for your work. So, you can compare the accuracy of the original estimates against the actual ones.

Learn more about how to estimate an issue

Project configuration

Your project configuration includes the configurations for issue screens that you should use to make the fields in the issue view editable if they were added through the board configurations.

> ℹ️ Any field, which you add by following these instructions, will be available and editable in the full issue view. However, the field won't appear on the issue open on the board or from the backlog until you add that field through the board configuration.

To open to the issue screen configurations:

1. In the upper-right corner of the screen, select **Administration** ⚙️ > **Projects**.
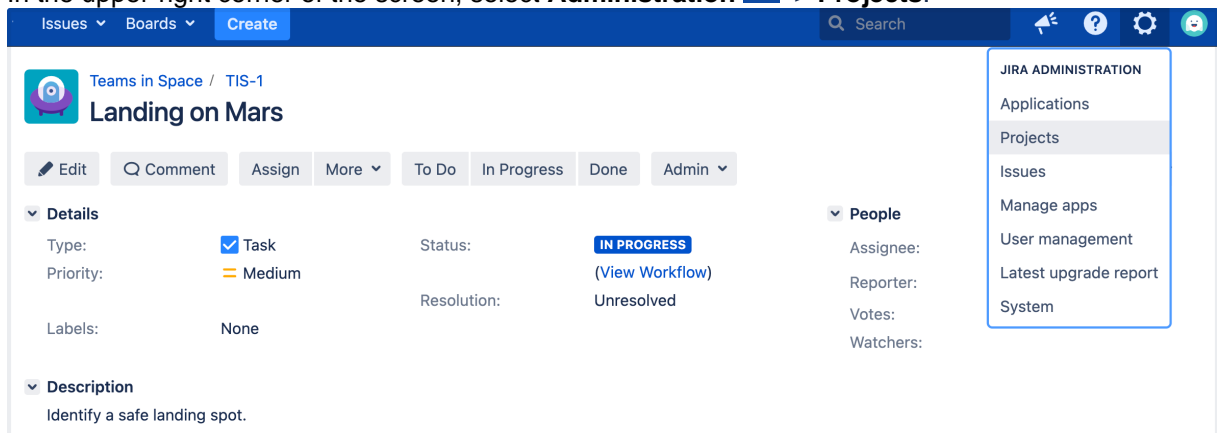
2. Select the project where you're customizing the view of issues.
3. Select **Screens**. Here you'll find the screen schemes for each issue type you have in the project.
4. Click on the scheme name to expand the details and make changes to the issue view.
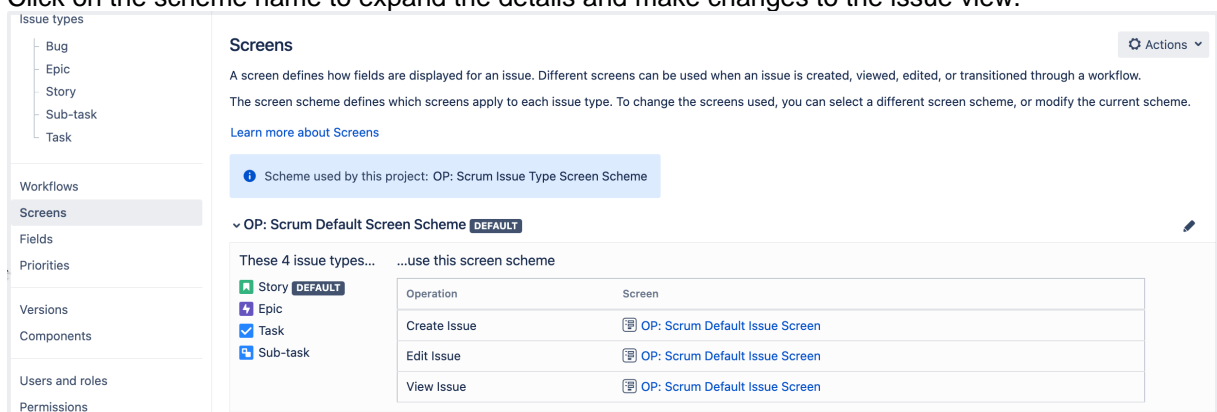
Making the added field editable in the issue view

To make the field editable, you should add it to the View Issue Screen with the help of the corresponding screen scheme:

1. Open the screen scheme for the **View Issue** operation. If all operations use the same screen scheme, your changes will be duplicated for the other operations.
2. Scroll down the list of fields in this scheme. You'll see the dropdown menu where you can find the field that you want to add and make editable.
3. Select **Add**.

4. Open the issue. You'll see that the **Bug Count** field is now editable.

> ℹ️ Some fields may not appear in the issue view immediately after you add them through the Screens configuration as they are required to have particular values. After you edit the issue and update the field's value, the field will appear in the full issue view.

Next steps

> ℹ️ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Configuring working days

The **Working Days** setting for your board is used for different reports and gadgets. You change this setting to filter out weekends, holidays, and other times during which your team might not be working on your board's project(s).

Before you begin

You must be a **Jira administrator** or a **board administrator** for the board to configure estimation or time tracking for it.

Configuring working days

1. Go to the desired board and select **Board** > **Configure**.
2. Click the **Working days** tab.
3. If your team has a different time zone from that of the server, select the team's **Time Zone**.
4. In **Standard Working Days**, select all the weekdays during which your team typically works.
5. In **Non-Working Days**, click **Add Date** to specify holidays or one-off dates during which your team will *not* be working.
   To remove a **Non-Working Day**, click the '**x**' next to the date.

These settings will be reflected in the following reports: Burndown Chart, Sprint Report, Epic Report, Version Report, Control Chart; and the following gadgets: Sprint Health Gadget, Burndown Gadget.

S*creenshot: the 'Board Configuration' screen — 'Working days' tab*

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

- Learn more about the charts that working days affect: Version Report, Control Chart, Burndown Chart, Sprint Report, Epic Report
- Learn how working days affect the Sprint Health Gadget, Burndown Gadget: Gadgets for Jira applications

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

Configuring development tools

Jira Software can be connected to a range of development tools to help you keep your project tracking in sync with your development work. Here are some of the features that you can unlock by connecting Jira Software to different tools:

- Connect to repository manager, like **Bitbucket** or **GitHub**, and you can configure issues to automatically transition when events occur in the repository, e.g commit and branches created, pull requests merged and declined, etc. You can also create branches directly from issues in Jira Software. In addition, every issue will show its related commits, branches, and pull requests.
- Connect to **Crucible** for code reviews, and you can configure issues to automatically transition when a review changes status. In addition, every issue will show its related reviews related to the issue.
- Connect to **Bamboo**, and every issue will show its related builds and deployments.

Connecting Jira Software to your development tools

Before you begin:

- Make sure that you have the right permissions. You need to be a Jira administrator to connect Jira Software to another application. You will need to be an administrator for your development tool as well.

Follow the instructions on Integrating with development tools *(Jira Admin documentation)* to connect Jira Software to your development tools.

Configuring Jira Software and your development tools

Some features will be automatically enabled when you connect Jira Software to your development tools. Others require some additional configuration. Each feature is listed below, including the configuration work required to set it up.

Development panel on issues
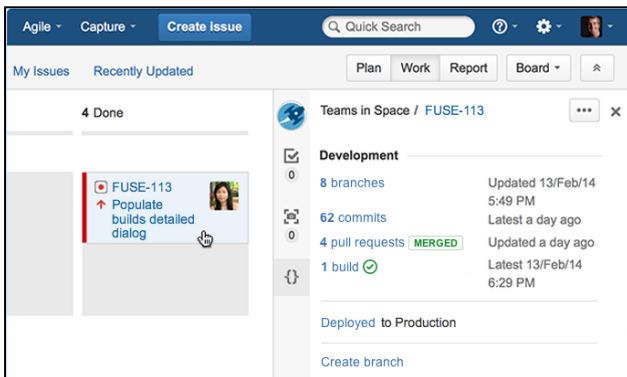
The **Development panel** is shown on every issue, and helps you to evaluate the development status of an issue at a glance. If you need to investigate further, click any item to display more details, or drill down into the development tool.

You don't need to do any further configuration once you have connected Jira Software to your development tools. However, your team will need to know how to reference issues in their development work.

Learn more: Referencing issues in your development work

View issue information in your developer tools

If you have connected Jira to one of the supported developer tools, like Bitbucket, Bamboo, Fisheye /Crucible, you will be able to view information about the issues in those tools. For example, in Bitbucket, you'll be able to see the issues that were referenced in a changeset; or in Bamboo, you'll be able to see the issues linked to a plan's builds.

Learn more: Bitbucket Server, Bitbucket Cloud, Bamboo, FishEye, Crucible

Create a branch from an issue

If you use Bitbucket Cloud or Bitbucket Server to manage your code repositories, you can create code branches directly from issues in Jira Software via the **Create branch** link. The link will open your connected repository application and launch the process of creating a branch for you. If you have multiple applications connected, then you can choose where you'd like to create the branch.
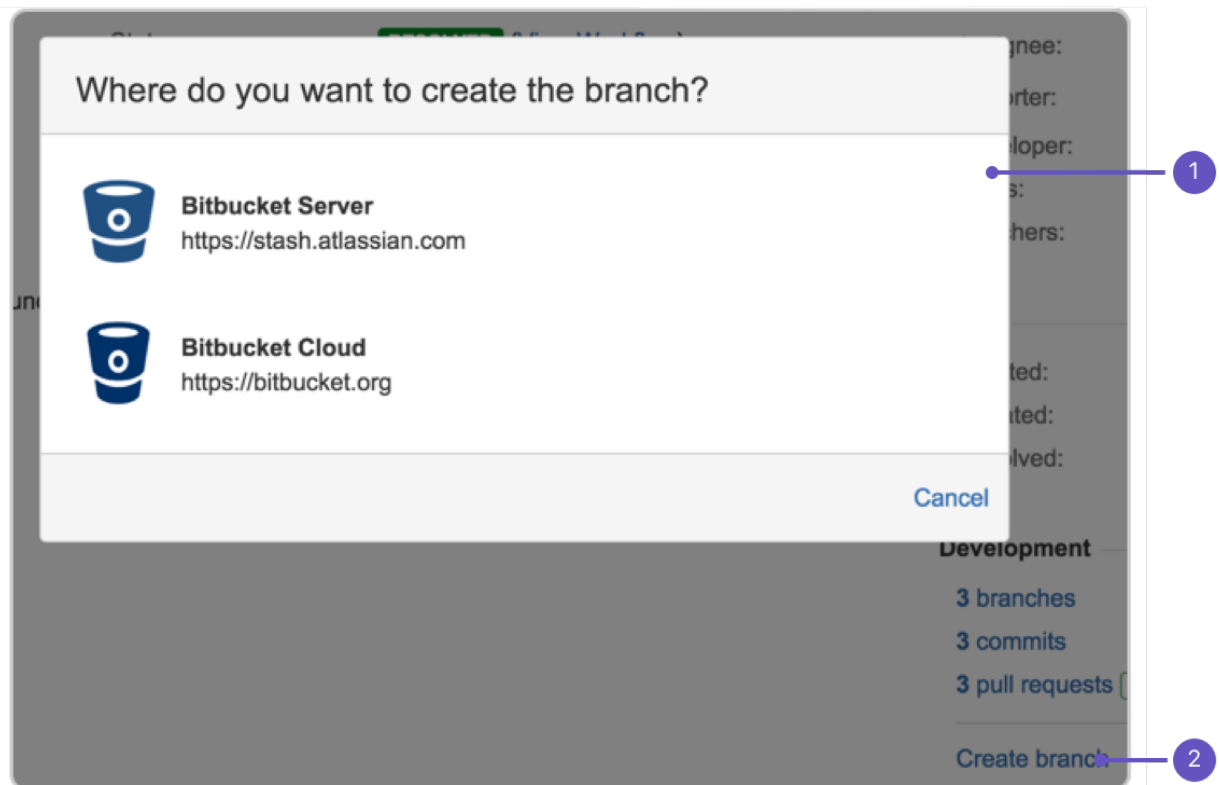
1. **Choose repo:** Select the repository to create the branch in.
2. **Create branch:** Click this link on your issue.

Workflow triggers
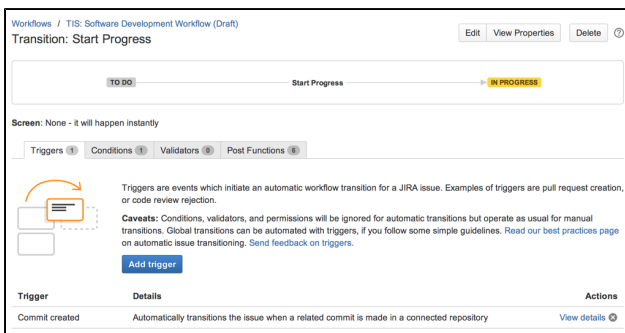
You can configure triggers in workflows that respond to events in your linked development tools. This helps your team keep their issue statuses in sync with their development work, rather than lagging behind. For example, when a developer creates a branch in Bitbucket to start work on an issue, they can reference the issue in the branch name, and the issue will automatically be transitioned from 'Open' to 'In progress'.

You will need to add triggers to your workflow(s) after connecting Jira Software to your development tools. We recommend that you read our guide on configuring workflow triggers.

Learn more: Configuring workflow triggers (Jira Admin documentation)

Development information in the Release Hub
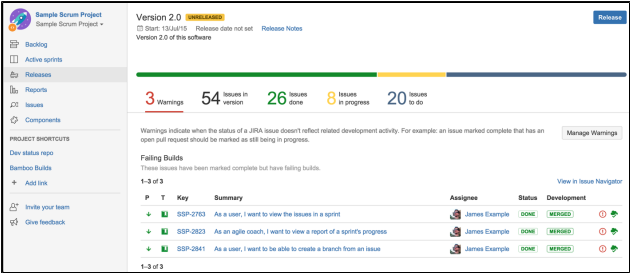
The Release Hub on your board lets you see the progress of your release (of a version) and determine what is likely to ship, at a glance. You can see all of the issues in the version, broken down by status category, as well as problems that could affect the release of the version (e.g. issues that should be in the release but are part of unmerged pulled requests, issues that haven't had code reviews, broken builds, etc).

You don't need to do any further configuration once you have connected Jira Software to your development tools. However, your team will need to know how to reference issues in their development work.

Learn more: Referencing issues in your development work

Development information in the list view of searches
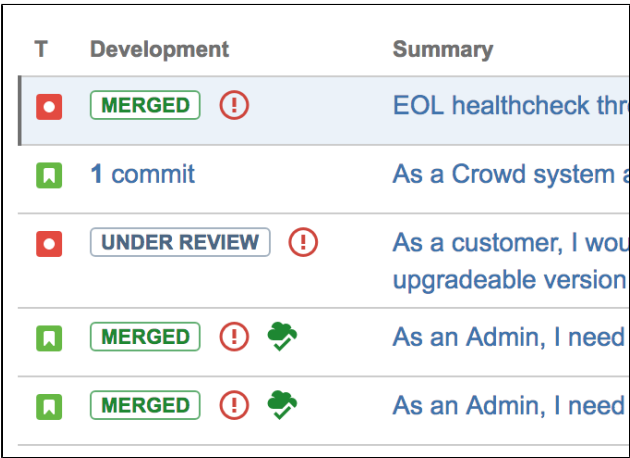
The **Development** column in your search list view shows you information from your connected development tools. If you need to investigate further, click any item to display more details, or drill down into the development tool.

Like the **Development panel**, you don't need to do any further configuration once you have connected Jira Software to your development tools, your team just needs to know how to reference issues in their development work.

Development information on your board cards

You can add development information to your board's cards so that it's available to you when you need it. This gives you greater context when viewing issues in a current sprint, or in your backlog.

To add Development information to your cards, navigate to you board and select **Board > Configure > Card layout**. Add the **Development** field to your cards.

Next steps

> (i) **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

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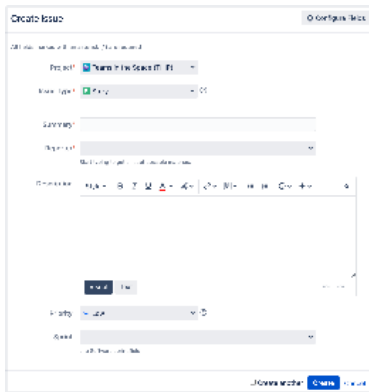
> ⓘ Jira administrators and project administrators have different permissions when it comes to customizing issues. Learn more about permissions in Jira

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 aren't 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.
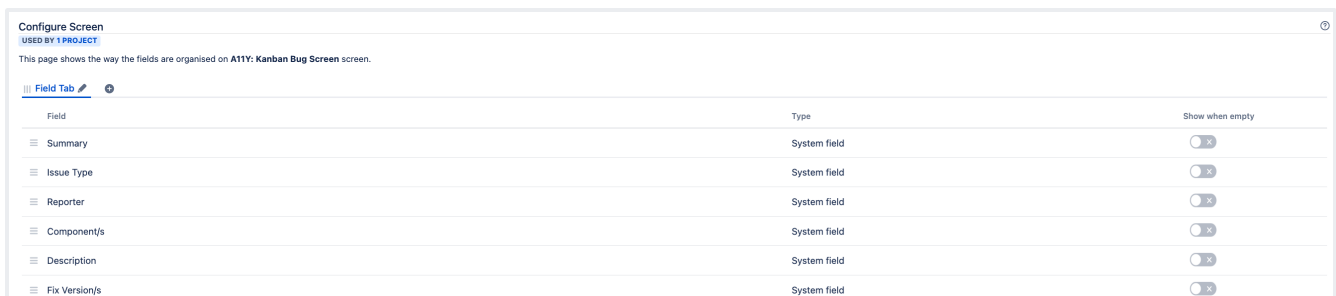
If your screen suits the preceding criteria, you can:

- 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**
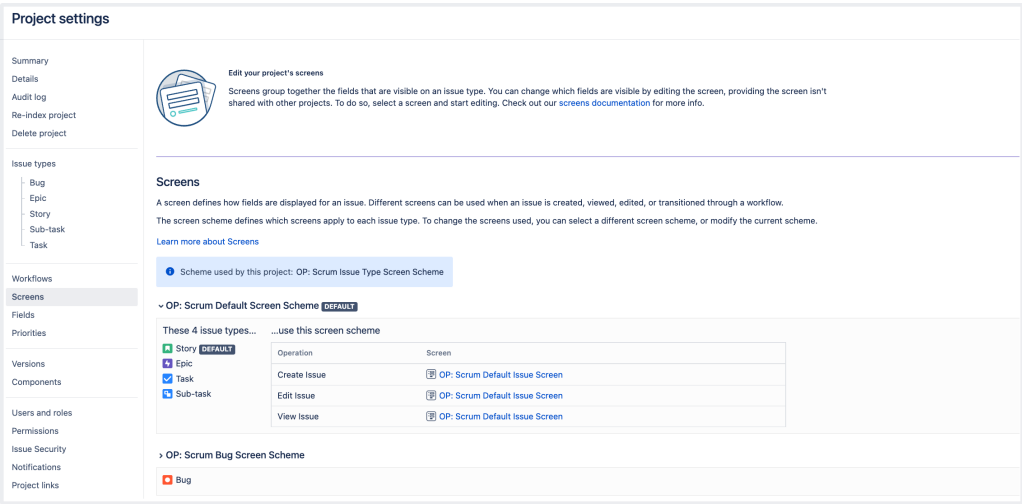
You'll see the screen schemes used by your project, and the issue types that use that scheme.

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.
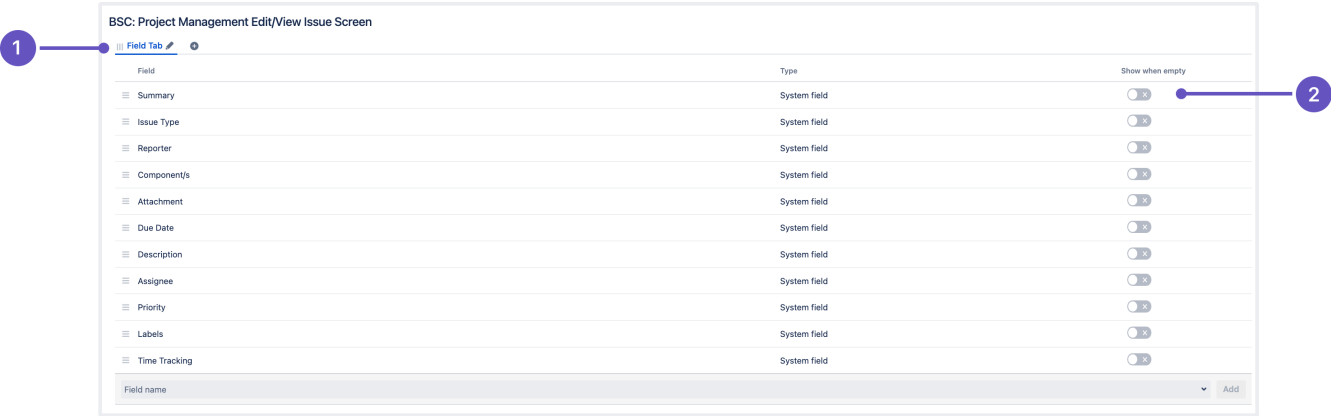
1. **Field tab** — this is how you can group related fields together.
2. **Show when empty** toggle — when enabled, empty custom fields will be visible in the issue view.

> ⓘ 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, you'll 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 tab's 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**.

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 information on customizing issues in the Jira administrator's documentation:

- Configuring issues
- Defining a screen

Working with issue type schemes

As a Jira admin, you can group different issue types into issue type schemes and associate these schemes with projects. This means that an issue type scheme associated with a project indicates which issue types are available in that project.

Jira admins can create, edit, copy, and delete issue type schemes, as well as assign them to projects. A project can be associated with only one issue type scheme.

Project admins can assign issue type schemes to their projects so that their teams use specific issue types to categorize work tasks in the projects.

To configure issue type schemes in your Jira project, go to **Administration** > **Issues**. Under **ISSUE TYPES**, select **Issue type schemes**.

What is an issue type scheme?

An issue type scheme is a group of issue types that become available in a project after you assign the scheme to the project. Only one issue type scheme can be assigned to a project. This provides your team with clear guidance on which issue type should be used for which task.

An issue type scheme allows you to:

- Set specific issue types for a project
- Set a default issue type for creating new issues
- Set a default order of issue types to select from when creating an issue

Default issue type scheme

All Jira instances come with the **Default Issue Type Scheme**. It groups issue types that are available in all projects on the instance unless the projects are associated with other schemes.

At the same time, each project that you can create in Jira has its own issue type scheme. But think of a project's issue type scheme as a template that you can edit, associate with other projects, or delete.

If you delete the project's issue type scheme, you can use the issue types from the project's default scheme.

The default scheme has the following characteristics:

- By default, it includes the following issue types: epic, story, and sub-task.

- When you create a new issue type, it'll be added to the default scheme automatically. You can reassign the new issue type to any other available scheme manually.
- When you create a new project that uses unique issue types, they won't be added to the default scheme automatically. For example, **Improvements** from the **Basic Software Project** won't appear in the default issue scheme. Also, you can't add this issue type manually to the default scheme. But you can assign it manually to other available issue type schemes.
- You can't unassign an issue type from the default scheme.
- You can't delete the default scheme.

Here's the default issue type scheme with the custom issue type "Design" added.

Adding a new issue type scheme

To add a new issue type scheme:

1. Go to **Administration** ⚙ > **System**.
2. Under **Issue types**, select **Issue type schemes**.
3. Select **Add Issue Type Scheme**.
4. Enter the name and description of the issue type scheme.
5. Drag the issue types that you want to have in the scheme from right to left. If needed, reorder them for the new scheme as you wish.
6. If needed, select a default issue type in the **Default Issue Type** field.

   ⓘ When you create a new issue in an associated project, the default issue type will be selected regardless of the order you set for the issue types in your scheme.

8. Select **Save**.

You can edit or copy your new issue type scheme at any time, as well as associate it with one of your projects.

Editing an issue type scheme

To edit an issue type scheme:

1. Go to **Administration** ⚙ > **Issues**.
2. Under **Issue types**, select **Issue type schemes**.
3. For the desired issue type scheme, select **Edit** in the **Actions** column.
4. Change the parameters of the scheme or the issue types that it includes.
5. Select **Save**.

Associating an issue type scheme with a project

To use issue types from a scheme, it must be associated with a project. Note that only one issue type scheme can be associated with each project.

To associate an issue type scheme with a project:

1. Go to **Administration** ⚙ > **Issues**.
2. Under **Issue types**, select **Issue type schemes**.
3. For the desired issue type scheme, select **Associate** in the **Actions** column.
4. Select a project that you want to associate with the issue type scheme.

5. Select **Associate**.

6. If you already have some issues in the project and their types aren't included in the selected issue type scheme, you'll see a list of these issues and a suggestion for what you should do with them. Select **Next**.

7. Map the unmatched issue types to the issue types from the scheme. Select **Next**.

8. If needed, select the fields where the original values should be retained. Select **Next**.

   The project you're associating with the issue type scheme may already be associated with a particular scr een scheme—a group of fields used by an issue type during a particular operation (creating, editing, or viewing.)

   This means that an issue type's fields and values in a particular project might not match the fields and values used by the same issue type scheme in another project.

   To avoid errors, you can choose to retain the original field values for issues of the matched types.

9. Check the summary of changes and select **Finish**.

Copying an issue type scheme

Copy an existing issue type scheme when you want to create a new scheme that should contain the required parameters. While copying, you'll be able to update the new scheme's parameters. Any projects associated with the original issue type scheme won't be copied to the new one.

To copy an issue type scheme:

1. Go to **Administration** ⚙️ > **Issues**.
2. Under **Issue types**, select **Issue type schemes**.
3. For the desired issue type scheme, select **Copy** in the **Actions** column.

You can now associate the new issue type scheme with one of your projects.

Deleting an issue type scheme

To delete an issue type scheme:

1. Go to **Administration** ⚙️ > **Issues**.
2. Under **Issue types**, select **Issue type schemes**.
3. For the desired issue type scheme, select **Delete** in the **Actions** column.
4. Confirm your action by selecting **Delete**.

Any project associated with the deleted scheme will now use issue types from the default issue type scheme.

Removing an issue type from an issue type scheme

To remove an issue type from a scheme:

1. On the **Issue type schemes** page, find a scheme from which you want to remove an issue type and select **Edit**.
2. Move an issue type that you want to exclude from the scheme out of the **Issue types for current scheme** column to the **Available issue types** column.

3. Select **Save**.

Modify Issue Type Scheme — SCRUM: Scrum Issue Type Scheme
USED BY 1 PROJECT

• ➕ Add issue type

Scheme Name*	SCRUM: Scrum Issue Type Scheme
Description	
Default Issue Type	Story ⌄

Change the order of the options by **dragging and dropping** the option into the desired order. Similarly, **drag and drop** the option from one list to the other to add or remove them.

Issue Types for Current Scheme

Remove all
- ✅ Task
- 🗃 Sub-task (sub-task)
- 📗 Story
- 🟥 Bug

Available Issue Types

Add all
- ⚡ Epic

Save   Reset   Cancel

After you select **Save**, you might see an error message stating that you can't remove the selected issue type from the scheme because the current project has archived issue that use this issue type.

If a project has archived issues that use the issue type you want to remove, you won't be able to remove the selected issue type.

The reason for this restriction is that archived issues are immutable so you can't edit them or migrate their issue type.

To get through this blocker, restore archived issues of the type you want to remove and proceed with the following steps in this instruction.

Learn more about archived issues and how to restore them

4. If a project has issues that use the issue type you want to remove from the scheme, you'll need to change the type of these issues to any other type that remains available in the issue type scheme. On the **Issue type migration** page, you'll see the overview of the changes that you should make. Select **Next** to proceed to the following step.

Issue Type Migration: Overview (Step 1 of 4)

Some issues have issue types that are no longer applicable. You will need to move these issues to another issue type. You will need to move each issue type for each project individually.

Original Project	Original Issue Type	Affected Issues
Scrum Development Project	📗 Story Created by Jira Software - do not edit or delete. Issue type for a user story.	17

Issue Type Migration: Overview
Step 1 of 4

<< Previous   Next >>   Cancel

5. Select an available issue type for affected issues to substitute the one you're removing.
6. Depending on the issue type you selected, you may need to set values for fields specific to this issue type. After you do this, select **Next**.
7. Check the settings for changing the issue type and if everything's correct, select **Finish**. The issue type migration will start.
8. After it completes, select **Ok, got it** to go back to your issue type schemes.

Configuring collaboration tools

Integrating collaboration tools with Jira Software can help you plan and run your project more efficiently. Connect to Hipchat (group and private chat), Confluence (wiki), or Team Calendars for Confluence to get issue notifications in your chat rooms, link project documentation to stories and epics, centralize your team leave, and more.

Connecting Jira Software to your collaboration tools

Follow the instructions on Integrating with collaboration tools *(Jira Admin documentation)* to connect Jira Software to Confluence and/or Hipchat.

Note:

- You need to be a Jira administrator to connect Jira Software to Confluence. You will need to be an administrator for Confluence as well.
- You need to be a Jira administrator or project administrator to connect Jira projects to Hipchat rooms.

Feature overview

If you have Jira Software, Confluence, and Team Calendars for Confluence, you can use them together to build stronger user stories and better plans for sprints or releases, while streamlining your team's work. Create meeting notes on the fly, wrap up a sprint with a retrospective report, and more.

Here are some of the features that you'll unlock by integrating Jira Software and Confluence. For a full feature overview, see Using Jira applications with Confluence.

Plan epics	If you have linked your Jira Software instance to a Confluence instance, you can create and link Confluence pages to your epics. For example, you may want to link your epic to a specification or design document in Confluence.
	Learn more...
Plan sprints	Team Calendars for Confluence acts as your single source of truth for managing team leaves, tracking projects, and planning events. If you have Team Calendars installed in Confluence, planning your sprints around leaves and events is even easier. Add a sprint event to your team's calendar in Confluence, and as you plan your sprints, you will be able to see whether team leaves or events will impact your work.
	Learn more...
Manage sprints	If you have linked your Jira instance to a Confluence instance, you can create and link Confluence pages to your sprints via Jira Software. For example, you may want to write up the sprint meeting notes in Confluence and link them to the sprint. You may also want to write your team's retrospective notes in Confluence and link them to the Sprint Report for the completed sprint.
	Learn more...
Manage releases	The release is underway. Provide your stakeholders with updates on how the release is going, by creating Jira Software status reports in Confluence. The Jira Software status report provides a snapshot of the release at a point in time. Stakeholders can monitor progress, even if they don't have access to Jira Software.
	Learn more...

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Organizing work with components

Components are used to group issues in a project, effectively creating sub-sections of a project. For example, you could set up components to group issues related to different workstreams, like the user interface or APIs. You can also set a default assignee for a component, which will cause all new issues to be assigned to that user. For example, you want all documentation issues to be assigned to your technical writer.

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".

Component Lead	The assignee will be set to the component leader.	If the component leader is not permitted to be assigned to issues in the permission scheme, this option will be disabled and will say "Component Lead is not allowed to be assigned issues". The Component Lead option will also not be available if the component does not have a lead assigned to the component. Instead, under this option, it will say "Component does not have a lead".
Unassigned	The assignee of the issue will not be set on the creation of this issue.	This option will only be available if "Allow unassigned issues" is enabled in the general configuration.

Editing a component's details

1. On the 'Components' screen, open the menu in the **Actions** column for the component you want to edit, and select **Edit**.
2. Edit the component's **Name**, **Description, Lead**, and **Default Assignee** in the **Edit component** dialog.
3. Click the **Save** button to save your changes.

Searching for a component

If you need to find a component in a long list, it's easiest to search for it. Start typing text into the search box that you know the component contains, and your list will automatically be filtered for you.

Deleting a component

1. On the 'Components' screen, open the menu in the **Actions** column for the component you want to delete, and select **Delete**.
2. You will be prompted to associate any issues assigned to this component with another component if you wish.
3. Click the **Delete** button to delete the component.

Building a backlog

You've set up your project and configured your systems. It's time to prepare the backlog of issues that your team will work from.

A backlog is simply a list of features, which could be for your product, service, project, etc. These features are not detailed specifications. Rather, they are usually described in form of user stories, which are short summaries of the functionality from a particular user's perspective. This is a common template for a user story: *As a <type of user>, I want <goal> so that I <receive benefit>*. For example, "As a developer, I want to log time against my tickets in the issue tracker during a sprint, so that I can show my progress."

Search the topics in 'Building a backlog':

Building a backlog in a Scrum project	In Scrum, it is critical to have a prioritized backlog, so the highest priority items are ready to be included in the next sprint. The **Backlog of your Scrum board** is designed to help you manage this. You can quickly add, update, and rank issues on the board to build the backlog. When it's time to start work, pushing issues from your backlog into sprints is as simple as dragging and dropping. Learn more: Using Plan Mode
Building a backlog in a Kanban project	In Kanban, you should also prioritize the backlog so that high priority items can be quickly picked up by your team. The **Backlog column of your Kanban board** lets you easily add issues, rank them, and categorize them. For example, you may use swimlanes to group high priority issues for great visibility, then rank the rest of the issues manually. If managing your backlog in the first column of your Kanban board becomes difficult, you may want to consider using the Kanban backlog. Learn more: Monitoring work in a Kanban project and Using your Kanban backlog

Already built a backlog?

Next: Plan a new version

Project lifecycle					
NEW PROJECT	BACKLOG	NEW VERSION	WORK	RELEASE	REPORTING

Using your Scrum backlog

The **Backlog** of a Scrum board shows the issues for your project(s) grouped into a backlog and sprints. In the Scrum backlog, you can create and update issues, drag and drop issues to rank them, or assign them to sprints, epics, or versions, manage epics, and more. You would typically use the Scrum backlog when building a backlog of issues, planning a new version, and planning a sprint.

Before you begin

Different functionalities in the Scrum backlog require different permissions. For example, to start a sprint, you need to be a project administrator for all projects that match the board's filter. See Permissions overview for more information.

About the Scrum backlog

An issue will only be visible in the Scrum backlog if:

- the issue is not a sub-task,
- the issue matches the board's saved filter,
- the issue's status maps to one of the board's columns (but not the **Done** column), and
- there is at least a status being mapped to the **right-most** column.
  *e.g. If you have the columns **To Do**, **In Progress**, and **Done**, ensure that you have a status mapped to **In Progress** at least. If you map all the statuses to the first column (**To Do**), you will not be able to see any issues in the **Scrum backlog**.*

Accessing the Scrum backlog

Go to the **Backlog** of your Scrum project.

Screenshot: the Scrum backlog (with an issue selected)
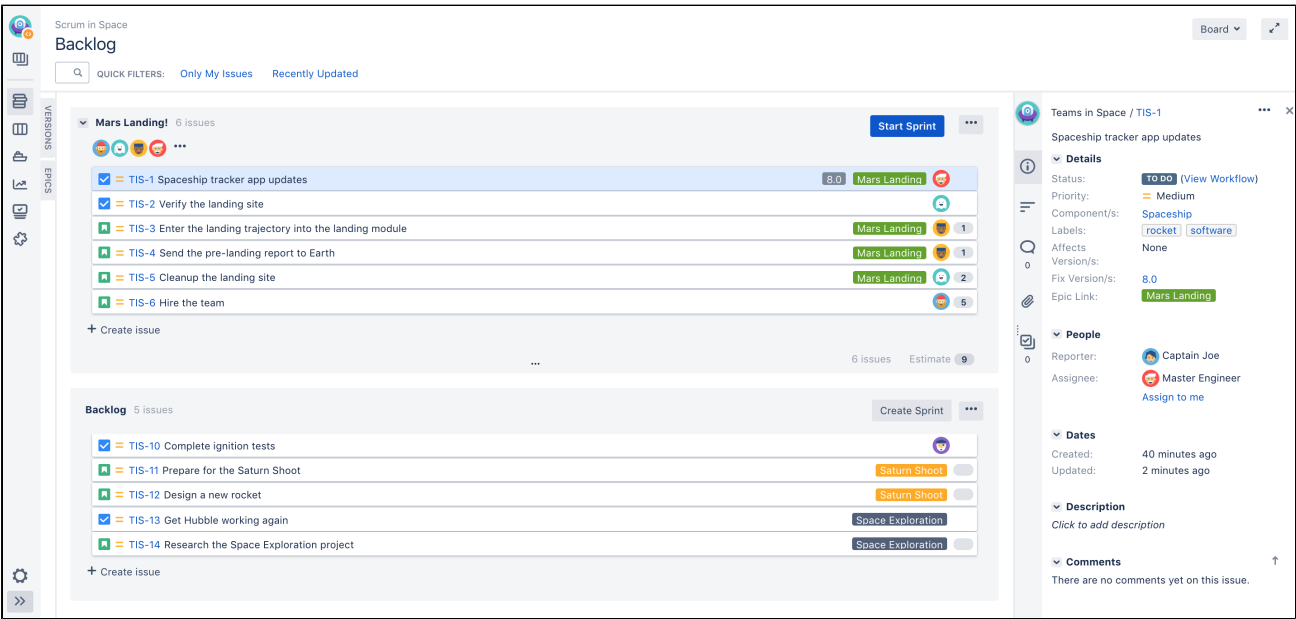
What can I do in the Backlog?

Task	Instructions

Add issues to the backlog	Click **Create** in the header to open the 'Create Issue' dialog and create your issue. The issue will be added to the backlog under the currently selected issue, or at the top of the backlog if no issue is selected. Tips: • Tick the **Create another** checkbox in the 'Create Issue' dialog to keep it open, if you are creating multiple issues. • You can quickly create issues using the inline issue create feature. Just click **+ Create issue**.
Prioritize the backlog	Drag and drop an issue to rank it. You can also right-click the issue to open a menu that allows you to send it to the top or the bottom of the backlog.
View and edit an issue's details	Click the desired issue on the board. The issue's details will display in the issue detail view, where you can also edit some issue details, depending on the issue detail view configuration. If you want to select an issue, rather than open the issue's details, use ctrl-left click or command-left click.
Estimate stories	Use the 'J' and 'K' keys to move through issues in the backlog and show the details on the right-hand side of the screen. Use the 'E' key to edit an issue, then update estimates or story points as you go. By default, the Story Points field is only available to issues of type 'Story' or 'Epic'. See Configuring estimation and tracking for details.
Identify the workload for specialists	The avatars for users (specialists) who have work assigned to them in a sprint are shown at the top of a sprint. Click ... (next to avatars) to view the sprint workload for assignees. If your Jira uses time tracking, you will also see the time required to complete this workload, shown in the Remaining Time Estimate column. To see the time required to complete the workload in a sprint, ask your admin to complete these steps: 1. Activate time tracking. 2. Enable time tracking for the board, and set the **Estimate Statistic** to anything but **Issue Count**.
Create sub-tasks	Select the desired issue, press the '.' key, and type 'create sub-task' to open the 'Create Sub-task' dialog. Create the sub-task as desired. Sub-tasks are useful for breaking a story (issue) down into implementable chunks.
Move an issue into a sprint	Drag and drop an issue from the **Backlog** section to the relevant **sprint** section, to move the issue from the backlog to the relevant sprint.
Flag an issue	Right-click an issue to open a menu and select **Add flag**. You can also add a comment with your flag, perhaps to indicate your reason for adding the flag.
Delete an issue	Right-click an issue and select **Delete**.
Find issues	Create your own Quick Filters to view only the issues you want. See Configuring Quick Filters for details. Search for an issue by name, code, or assignee.

Organize how many issues are visible	We want to make sure you don't have to wait for your backlog to load, so, by default, we show you a maximum of 100 issues (90 from the top of your backlog and 10 from the bottom) and hide the remaining ones. If you want to see all issues, just click the **Show all issues** hyperlink that splits your backlog. Then, we'll load all your issues.
	If you have more than 100 issues, you can also click **More** (⋯) and select how many you want to display. You can select to see 100, 500 or all issues.
Organize stories into epics	Click **EPICS** (aligned vertically, left side of the board) to show the 'EPICS' panel. You can create epics, drag and drop issues into epics, and filter by epics via this panel.
	An epic is essentially a large user story, used for grouping smaller stories. For example, you may have a 'Performance' theme for your release, which you could capture as an epic.
Plan versions	Click **VERSIONS** (aligned vertically, left side of the board) to show the 'VERSIONS' panel. You can create and edit versions, assign issues to versions via drag-and-drop, and filter by versions via this panel.
	Note, you need to have at least one version in your project for this panel to display.
Plan sprints	Click **Create Sprint**, then drag and drop issues into your new sprint. You can also drag and drop the horizontal divider to add or remove multiple issues. The sprint footer will display the number of issues and total estimated work.
	When you're happy with the issues for the sprint, click **Start Sprint**, and the stories will move into the **Active sprints**.
	While a sprint is active in the Active sprints, you can still plan subsequent sprints in the Backlog, but you won't be able to start them until the active sprint is completed. (You can, however, drag and drop an issue in the Backlog onto the active sprint.)
Add goals to sprints	Click **More** (⋯) **> Add sprint goal**, then enter a goal for this sprint to keep your team focused. The sprint goal will be displayed below the sprint name in the backlog, scrum board, and sprint report.
Delete inactive sprints	If you haven't started the sprint yet, you can delete it by clicking **More** (⋯) **> Delete sprint**. You can also delete completed sprints in **Reports > Sprint Report**.
Hide issue detail view	If you're switching between the issues on the list, it might be useful to hide the issue detail view that appears on the right of the backlog. You can do it by clicking **More** (⋯) **> Hide detail view**.

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Running sprints in a Scrum project
- Ranking an issue
- Estimating an issue
- Working with epics

123

Using your Kanban backlog

Before you begin

- You can only use the Kanban backlog if it's already enabled by a **Jira administrator** or a **board administrator**.
- The Kanban backlog *must be enabled for a particular board*, for the board users to use it. See Enabling the Kanban backlog for more information.

About the Kanban backlog

Managing your backlog in the first column of your Kanban board is easy to do — as long as there are only a few issues in your backlog. As your backlog grows, viewing and scrolling through these issues can become difficult.

If you're planning work for your team, you get a bigger backlog that gives you an optimized list view of the issues you're creating and ranking for your team. If you're a team member working on a Kanban project, you can focus on your work-in-progress on the Kanban board, without the distraction of items in planning.

Accessing the Kanban backlog

Navigate to the **Backlog** of your Kanban software development project.

*Screenshot: sample Kanban backlog (with an issue selected), with **Backlog** and **Selected for Development** as board columns*
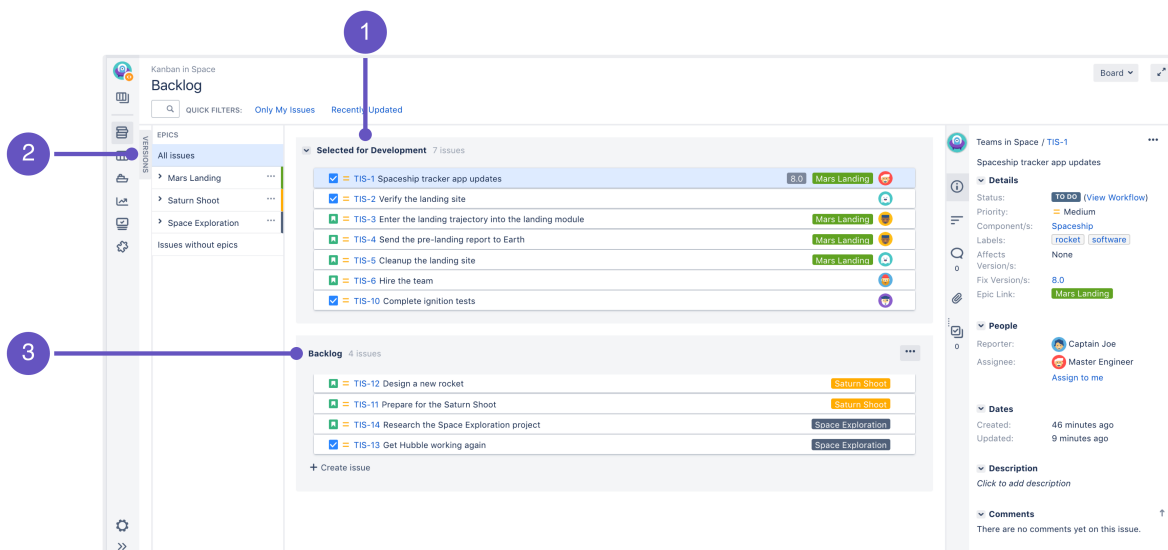
1. **Selected for Development:** Issues in the **Selected for Development** column of the Kanban board.
2. **Versions** (hidden) and **epics** (expanded) panels. Select your versions or epics to see the associated issues, or drag your issues onto them to make the associations.
3. **Backlog:** Issues in the **Kanban backlog** column of the Kanban board.

Note in the screenshot above, the Kanban backlog shows issues in both **Backlog** and **Selected for Development** sections. This makes it easy for you to move issues from the **backlog** to **selected for development**. Issues selected for development will then appear in your Kanban board, and your team can start working on them accordingly.

An issue will only be visible in the Kanban backlog if:

- the issue matches the board's saved filter, and
- the issue's status is in the **Kanban backlog column**, or the **next column** on the Kanban board.

What can I do in the Kanban backlog?

Task	Instructions
Add issues to the backlog	Click **Create** in the header to open the 'Create Issue' dialog and create your issue. The issue will be added to the backlog under the currently selected issue, or at the top of the backlog if no issue is selected. Tips: - Tick the **Create another** checkbox in the 'Create Issue' dialog to keep it open, if you are creating multiple issues. - You can quickly create issues using the inline issue create feature. Just click **+ Create issue**.
Prioritize the backlog	Drag and drop an issue to rank it. You can also right-click the issue to open a menu that allows you to send it to the top or the bottom of the backlog.
View and edit an issue's details	Click the desired issue. The issue's details will display in the issue detail view, where you can also edit some issue details, depending on the issue detail view configuration. To open the issue in a separate tab or window, right-click on the issue key. To view the details of an epic issue, click the epic lozenge on the issue itself.
Create sub-tasks	Click the desired issue. Select **More** (...) **> Create sub-task** to open the 'Create sub-task' screen. Create the sub-task as desired. Sub-tasks are useful for breaking a story (issue) down into implementable chunks. - A newly created sub-task is displayed with its parent task expanded. The sub-task itself is slightly indented after its parent task. - You can view sub-tasks by expanding their parent tasks. - Sub-tasks may or may not be included in the issue count. This depends on column constraints configured for the board.

Transitio n an issue	Drag and drop an issue from the **Backlog** section to the **Selected for Development** section (or the next column of your Kanban board), to move the issue from the backlog to the next status in your workflow. Note, the **Selected for Development** section may be named differently, depending on the colu mn configuration of your Kanban board. Depending on the following conditions, you can transition multiple issues from the backlog to the next status in your project's workflow: • You have the transition issues permission. • The statuses of the issues are already mapped to the board's columns. • Your project's workflow supports the particular transition for the multiple issues selected, and all transition conditions are met. • The issues you're transitioning have the same set of transitions. • The issues you're transitioning don't have any associated transition screen. • You can select multiple issues by using the **Shift** or **Ctrl** key while clicking the issues. However, you can't transition multiple issues by dragging them to another column. You can only move multiple issues within the same column. • To transition both the parent issue and its sub-tasks, drag and drop the parent issue while it's collapsed to the corresponding section. • To transition just the parent issue or a sub-task, expand the parent issue first, then drag and drop only the parent issue or the sub-task to the corresponding section. If your **Selected for Development** column has two statuses, one with a transition screen and another without it, your issues will be assigned to the status that doesn't require a transition screen. See Transitioning an issue for more information.
Flag an issue	Right-click an issue and select **Add flag**. You can also add a comment with your flag, perhaps to indicate your reason for adding the flag.
Delete an issue	Right-click an issue and select **Delete**.
Find issues	Create your own Quick Filters to view only the issues you want. See Configuring Quick Filters for details. Search for an issue by name, code, or assignee.
Decide how many issues are visible	We want to make sure you don't have to wait for your backlog to load, so, by default, we show you a maximum of 100 issues (90 from the top of your backlog and 10 from the bottom) and hide the remaining ones. If you want to see all issues, just click the **Show all issues** hyperlink that splits your backlog. Then, we'll load all your issues. If you have more than 100 issues, you can also click **More** (···) and select how many you want to display. You can select to see 100, 500 or all issues.
Select issues for your team to work on	After grooming the issues in your backlog, you can select which issues your team will start working on, by dragging and dropping issues from the Backlog section to the Selected for Development section. Issues selected for development will then appear in your Kanban board, and your team can start working on them accordingly. Note that you can move a sub-task from the Backlog to Selected for Development independently of its parent, and vice versa.

Organize stories into epics	Click **EPICS** (aligned vertically, left side of the board) to show the 'EPICS' panel. You can create epics, drag and drop issues into epics, and filter by epics via this panel. An epic is essentially a large user story, used for grouping smaller stories. For example, you may have a 'Performance' theme for your release, which you could capture as an epic.
Plan versions	Click **VERSIONS** (aligned vertically, left side of the board) to show the 'VERSIONS' panel. You can create and edit versions, assign issues to versions via drag-and-drop, and filter by versions via this panel. Note, you need to have at least one version in your project for this panel to display.
Hide issue detail view	If you're switching between the issues on the list, it might be useful to hide the issue detail view that appears on the right of the backlog. You can do it by clicking **More** (⋯) **> Hide detail view**.

Next steps

ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Create an issue and a sub-task
- Configuring columns
- Disabling the epics panel
- Transitioning an issue
- Configuring versions
- Managing epics

Planning a version

You've got a backlog full of exciting stories that are just waiting to be turned into an amazing application. It's time to plan a version!

The documentation in this section will help you set up a new version for your Jira Software project. This includes creating the version, setting up epics, and assigning work to a version.

Search the topics in 'Planning a version':

Overview

Creating and configuring a version	A version is a set of features and fixes released together as a single update to your application. Assigning issues to versions helps you plan the order in which new features (stories) for your application will be released to your customers.
	Learn more: Configuring versions in a Scrum project and Configuring versions in a Kanban project
Working with epics *(Scrum boards only)*	An *epic* captures a large body of work. It is essentially a large user story that can be broken down into a number of smaller stories. It may take several sprints to complete an epic. You might use epics to capture broader themes in a release, e.g. performance-related work.
	Learn more: Working with epics
Branching your repository	You can use a number of different branching strategies, including task branching, feature branching, and release branching. You'll need to decide how much work should to be contained in a branch before merging, when deciding on a strategy.
	If you decide to use the release branching model, you'll create your branches at the start of a version. For example, you might create a branch for your last version, so you can continue with bugfix work on that branch, while new development work is done on the master branch.
	Learn more: Branches (Bitbucket Server documentation), Branching a repository (Bitbucket Cloud documentation), Creating and deleting branches (GitHub documentation)
	Tip: You can create a branch via an issue, if Jira Software is connected to Bitbucket Cloud, Bitbucket Server, or GitHub.

Already planned a version?

Next: Get to work!

Project lifecycle					
NEW PROJECT	BACKLOG	NEW VERSION	WORK	RELEASE	REPORTING

Configuring versions in a Scrum project

A version is a set of features and fixes released together as a single update to your application. Assigning issues to versions helps you plan the order in which new features (stories) for your application will be released to your customers. In Jira Software, you can view your issues according to which version they belong to. You can also drag-and-drop issues into a version to assign them to it, which you should do before you start work on the issues. This helps you plan your upcoming versions, which may span multiple sprints.

On this page:

- Before you begin
- Managing versions
- Tips
- Next steps

Before you begin

- This page does not apply to Kanban projects. See Configuring versions in a Kanban project instead.
- The functions for managing versions require different permissions. For example, you need the 'Edit Issues' permission to add an issue to a version. Read Permissions overview for more information.

Managing versions

1. Navigate to your board.
2. Click **Backlog**. If the Versions panel is not showing, click **VERSIONS** on the left side of the board (aligned vertically) to open it.

Add a new version	Click **Create version** (you will need to hover over the 'VERSIONS' panel to show this link), enter the version details, and create it. * The **Start Date** is used to give you a more accurate Version Report in cases where you might plan a version many weeks or even months in advance, but not actually commence work until closer to the release date. * The **End Date** is used to calculate the days remaining in a release on the Release Hub.
Update a version's details	For the version name, click the arrow next to the name, then choose **Edit name**. For other fields (e.g. Description), click the field to edit it.
Add an issue to a version	Drag and drop the issue onto the version in the 'VERSIONS' panel.
Remove an issue from a version	Drag and drop the issue onto **Issues without versions** in the 'VERSIONS' panel.
Filter issues by version	Click the version in the 'VERSIONS' panel to show only issues in that version. Click **All issues** to remove the filter. *Alternatively, click **Clear all filters** next to the sprint's name.*

Screenshot: viewing versions in the Backlog
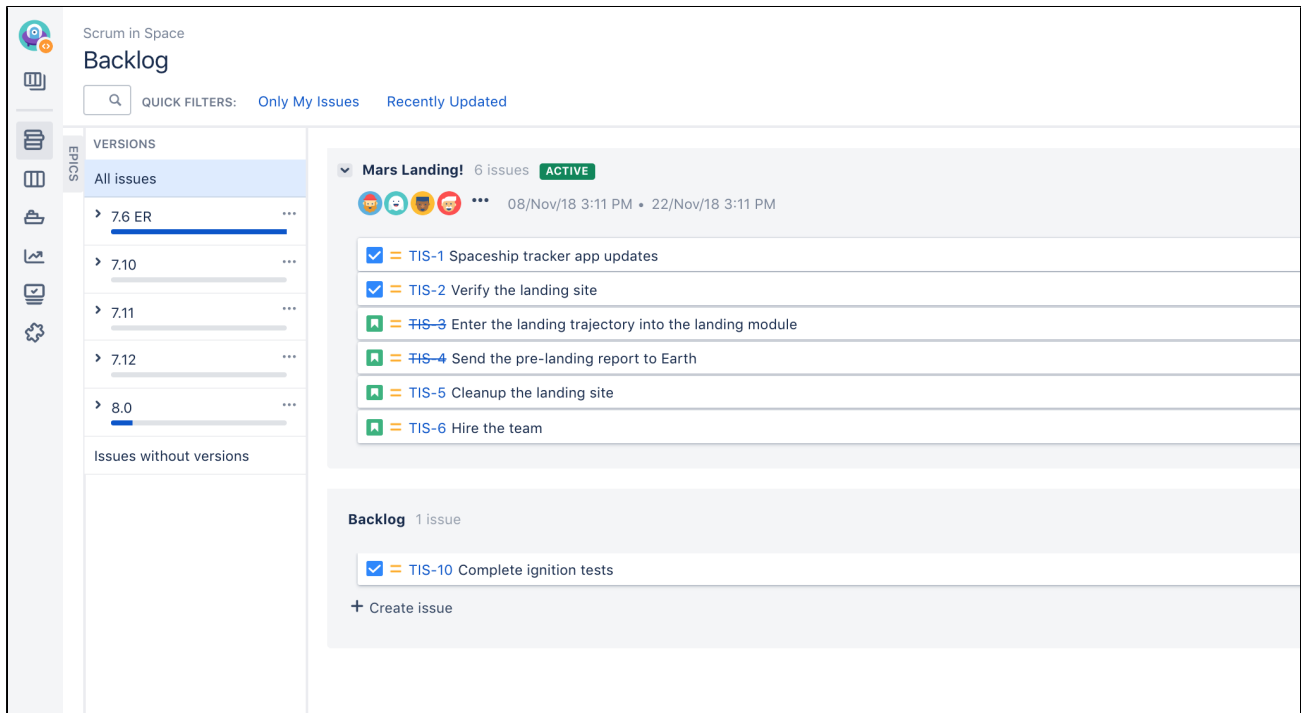
Tips

- The blue horizontal bar under the version's name (in the 'VERSIONS' panel) indicates progress towards completing the work estimated for the version. Note that this bar only represents progress on the estimated issues in a version.
- You can use the JQL `fixVersion` field to search for a version's issues — see JQL *(Jira Admin documentation)* for details.

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Using your Scrum backlog
- Version Report
- Deploying a release

Configuring versions in a Kanban project

A version is a set of features and fixes released together as a single update to your application. Assigning issues to versions helps you plan the order in which new features (stories) for your application will be released to your customers.In Jira Software, you can view your issues according to which version they belong to. You can also drag-and-drop issues into a version to assign them to it, which you should do before you start work on the issues. This helps you plan your upcoming versions, which may span multiple sprints.

On this page:

- Before you begin
- Managing versions
- Tips
- Next steps

Before you begin

The functions for managing versions require different permissions. For example, you need the 'Edit Issues' permission to add an issue to a version. Read Permissions overview for more information.

Managing versions

1. Navigate to your board.
2. Click **Backlog**. If the Versions panel is not showing, click **VERSIONS** on the left side of the board (aligned vertically) to open it.

Add a new version	Click **Create version** (you will need to hover over the 'VERSIONS' panel to show this link), enter the version details, and create it. * The **Start Date** is used to give you a more accurate Version Report in cases where you might plan a version many weeks or even months in advance, but not actually commence work until closer to the release date. * The **End Date** is used to calculate the days remaining in a release on the Release Hub.
Update a version's details	For the version name, click the arrow next to the name, then choose **Edit name**. For other fields (e.g. Description), click the field to edit it.
Add an issue to a version	Drag and drop the issue onto the version in the 'VERSIONS' panel.
Remove an issue from a version	Drag and drop the issue onto **Issues without versions** in the 'VERSIONS' panel.
Filter issues by version	Click the version in the 'VERSIONS' panel to show only issues in that version. Click **All issues** to remove the filter. *Alternatively, click **Clear all filters** next to the column's name.*

Screenshot: viewing versions in the Backlog

Tips

- You can use the JQL `fixVersion` field to search for a version's issues — see JQL *(Jira Admin documentation)* for details.

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Using your Scrum backlog
- Version Report
- Deploying a release

Working with epics

An epic captures a large body of work, e.g. performance-related work in a release. It is essentially a large user story that can be broken down into a number of smaller stories. An epic can span more than one project, if multiple projects are included in the board where the epic is created.

Unlike sprints, scope change in epics is a natural aspect of agile development. Epics are almost always delivered over a set of sprints. As a team learns more about an epic through development and customer feedback, some user stories will be added and removed to optimize the team's release time.

Depending on whether you use Scrum or Kanban, see Managing epics in a Scrum project, or Managing epics in a Kanban project. In general

The way epics are displayed and configured between Scrum and Kanban boards is mostly the same. See Managing epics in a Scrum project, or Managing epics in a Kanban project for more information.

On this page:

- Linking a Confluence page to an epic
- Next steps

Linking a Confluence page to an epic

If you have linked your Jira Software instance to a Confluence instance, you can create and link Confluence pages to your epics. For example, you may want to link your epic to a specification or design document in Confluence.

Learn more: Linking a Confluence page to an epic

Next steps

ℹ️ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Managing epics in a Scrum project
- Managing epics in a Kanban project
- Epic Report

Managing epics in a Scrum project

Before you begin

The functions for managing epics require different permissions. For example, you need the 'Edit Issues' permission to add an issue to an epic. Read Permissions overview for more information.

Managing epics

1. Go to your Scrum backlog.
2. Click **EPICS** on the left side of the board (aligned vertically) to open it.

Add a new epic	Click **Create epic** (you will need to hover over the 'EPICS' panel to show this link), enter the epic details, and create it.
Update an epic's details	For the epic name, click the arrow next to the name, then choose **Edit name**. For other fields, click the arrow next to the epic's name, then choose **View epic details**. You can then edit the epic like any other issue.
Change an epic's color on the board	Click the arrow next to the epic's name, then choose the color from the menu.
Add an issue to an epic	To add an existing issue, drag and drop the issue onto the epic in the 'EPICS' panel. To add a new issue, click **Create issue in epic** (if this does not show, you need to expand the epic details in the 'EPICS' panel.
Remove an issue from an epic	Drag and drop the issue onto **Issues without epics** in the 'EPICS' panel. *Alternatively, view the detailed view of the issue on the board, locate the **Epic** field, then click the 'x' in the epic name lozenge (this method also works in the Active sprints of a board).*
Filter issues by epic	Click the epic in the 'EPICS' panel to show only issues in that epic. Click **All issues** to remove the filter. *Alternatively, click **Clear all filters** next to the sprint's name.*
Complete an epic	Click the arrow next to the epic's name, then choose **Mark as Done** from the menu. *This will set the epic's **Epic Status** field to "Done", but will not affect the epic's workflow or its **Status** field , and n one of the epic's issues will be affected.*

Screenshot: viewing an epic in a Scrum backlog
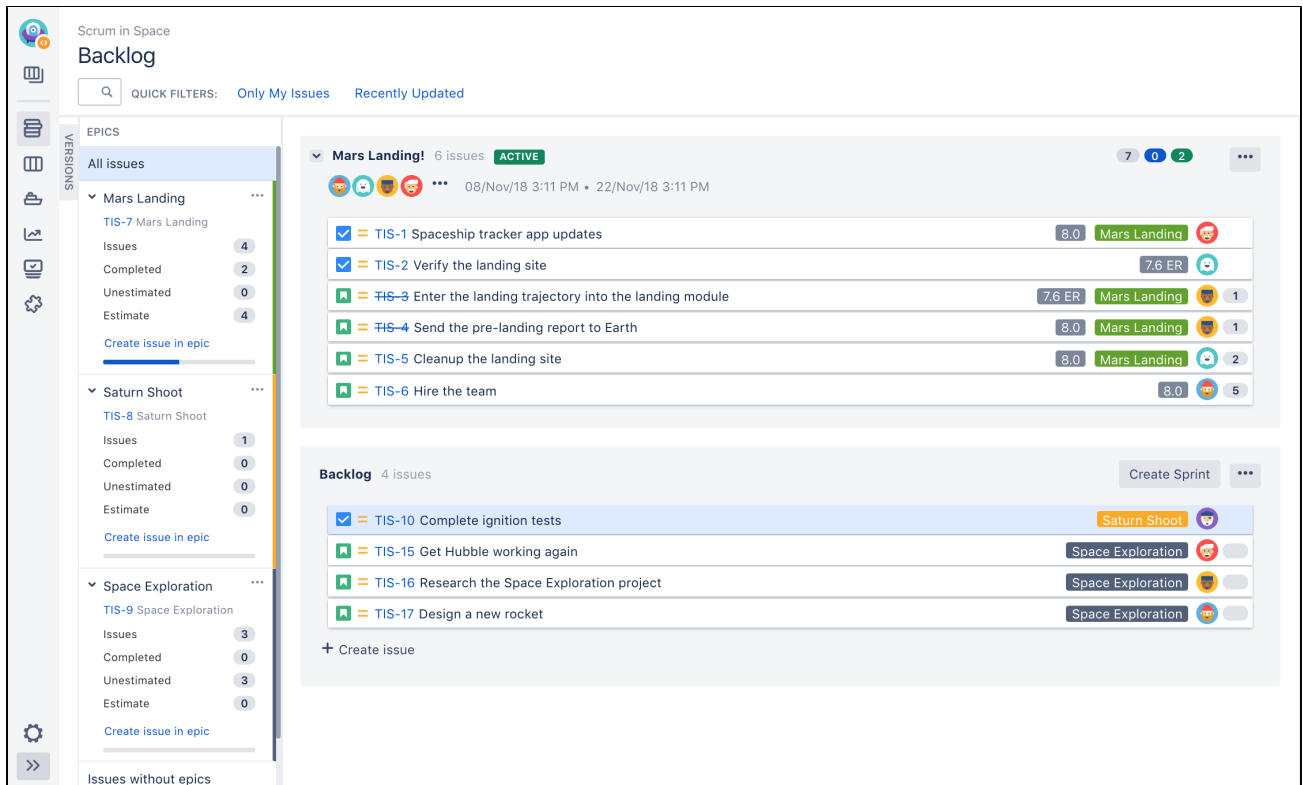
Note that the blue horizontal bar under an epic's details (in the 'EPICS' panel) indicates progress towards completing the work estimated for the issues added to the epic. This bar only represents progress on the estimated issues in an epic.

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Using your Scrum backlog
- Epic Report

135

Managing epics in a Kanban project

Before you begin

- The functions for managing epics require different permissions. For example, you need the 'Edit Issues' permission to add an issue to an epic. Read Permissions overview for more information.
- The epics panel must be enabled for your board. It's enabled by default, so you should see it unless your board administrator has disabled it.

Managing epics

1. Go to your Kanban backlog.
2. Click **EPICS** on the left side of the board (aligned vertically) to open it.

Add a new epic	Click **Create epic** (you will need to hover over the 'EPICS' panel to show this link), enter the epic details, and create it.
Update an epic's details	For the epic name, click the arrow next to the name, then choose **Edit name**. For other fields, choose **View epic details** . You can then edit the epic like any other issue.
Change an epic's color on the board	Click the arrow next to the epic's name, then choose the color from the menu.
Add an issue to an epic	To add an existing issue, drag and drop the issue onto the epic in the 'EPICS' panel. To add a new issue, click **Create issue in epic** (if this does not show, you need to expand the epic details in the 'EPICS' panel.
Remove an issue from an epic	Drag and drop the issue onto **Issues without epics** in the 'EPICS' panel. *Alternatively, view the detailed view of the issue on the board, locate the **Epic** field, then click the 'x' in the epic name lozenge (this method also works in the Active sprints of a board).*
Filter issues by epic	Click the epic in the 'EPICS' panel to show only issues in that epic. Click **All issues** to remove the filter. *Alternatively, click **Clear all filters** next to the sprint's name.*
Complete an epic	Click the arrow next to the epic's name, then choose **Mark as Done** from the menu. *This will set the epic's **Epic Status** field to "Done", but will not affect the epic's workflow or its **Status** field , and n one of the epic's issues will be affected.*

Screenshot: viewing an epic in a Kanban backlog

Next steps

> ℹ️ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Using your Kanban backlog
- Epic Report

Linking a Confluence page to an epic

If you have linked your Jira Software instance to a Confluence instance, you can create and link Confluence pages to your epics. For example, you may want to link your epic to a specification or design document in Confluence.

Screenshot: linked pages for an epic (Scrum board)
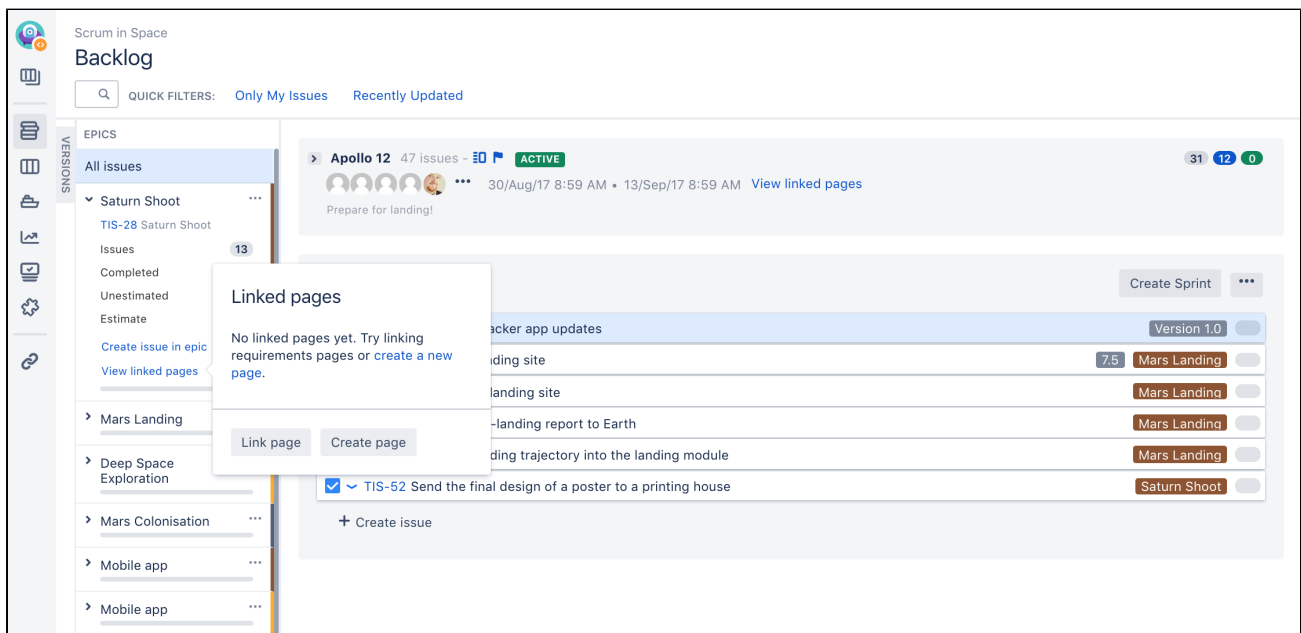
Creating a new linked Confluence page for an epic

Note, these instructions only apply to Scrum boards.

1. Log in to Jira.
2. Navigate to your desired board.
3. Click **Backlog** and click **EPICS** (aligned vertically, left side of the board) to show the 'EPICS' panel.
4. Click **Linked pages** at the bottom of the epic details in the **Epics** panel.
5. Click **Create page**. The 'Create' dialog in Confluence will be displayed, with the Product Requirements blueprint selected (select a different blueprint, if you don't want to use it).

Note, you cannot create a new linked Confluence page when viewing an epic in Jira Software, although you can link an existing Confluence page (see instructions below).

Linking an existing Confluence page to an epic

... in Backlog *(Scrum boards only)* **:**

1. Log in to Jira.
2. Navigate to your desired board.
3. Click **Backlog** and click **EPICS** (aligned vertically, left side of the board) to show the 'EPICS' panel.
4. Click **Linked pages** at the bottom of the epic details in the **Epics** panel.
5. Click **Link page**, search for the page you want, and select it.

... when viewing the epic in Jira Software   *(Scrum and Kanban boards)* **:**

1. *View the issue or story in Jira. Note, you need to view the issue in its own browser tab, i.e. not in the issue detail view on the board.*
2. *Click **More** > **Link**. This displays the Link dialog.*
3. *Click the **Confluence Page** tab in the dialog, and enter the details for the page you want to link to.*

Next steps

ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read these related topics:

- Using JIRA applications with Confluence
- Working with epics

Working with epic statuses

To manage different layers of your project work, there are a few status fields in Jira. You can have Epic statues, Issues statuses and Workflow statuses.

Epic statuses

Since issues in epics can be from multiple projects and the issues contained in the epic might have different workflows and statuses, an epic status is different from the workflow status. Epic status has three values by default:

- To Do
- In Progress
- Done

> ⓘ The epics displayed in the Epics panel have one of the following statuses: To Do, or In Progress.

The epic status field is hidden by default.

Marking an epic as Done

Epic status isn't an automatic custom field, and you can set the value manually. If you set the Epic to Done, it will disappear from the agile board and you will not be able to find it in the **Epic Link** field. It will also disappear from the **Epics** panel.

You can mark your epic as done by going to **Backlog**, opening the **Epics** panel and selecting **Mark as Done** for your epic.
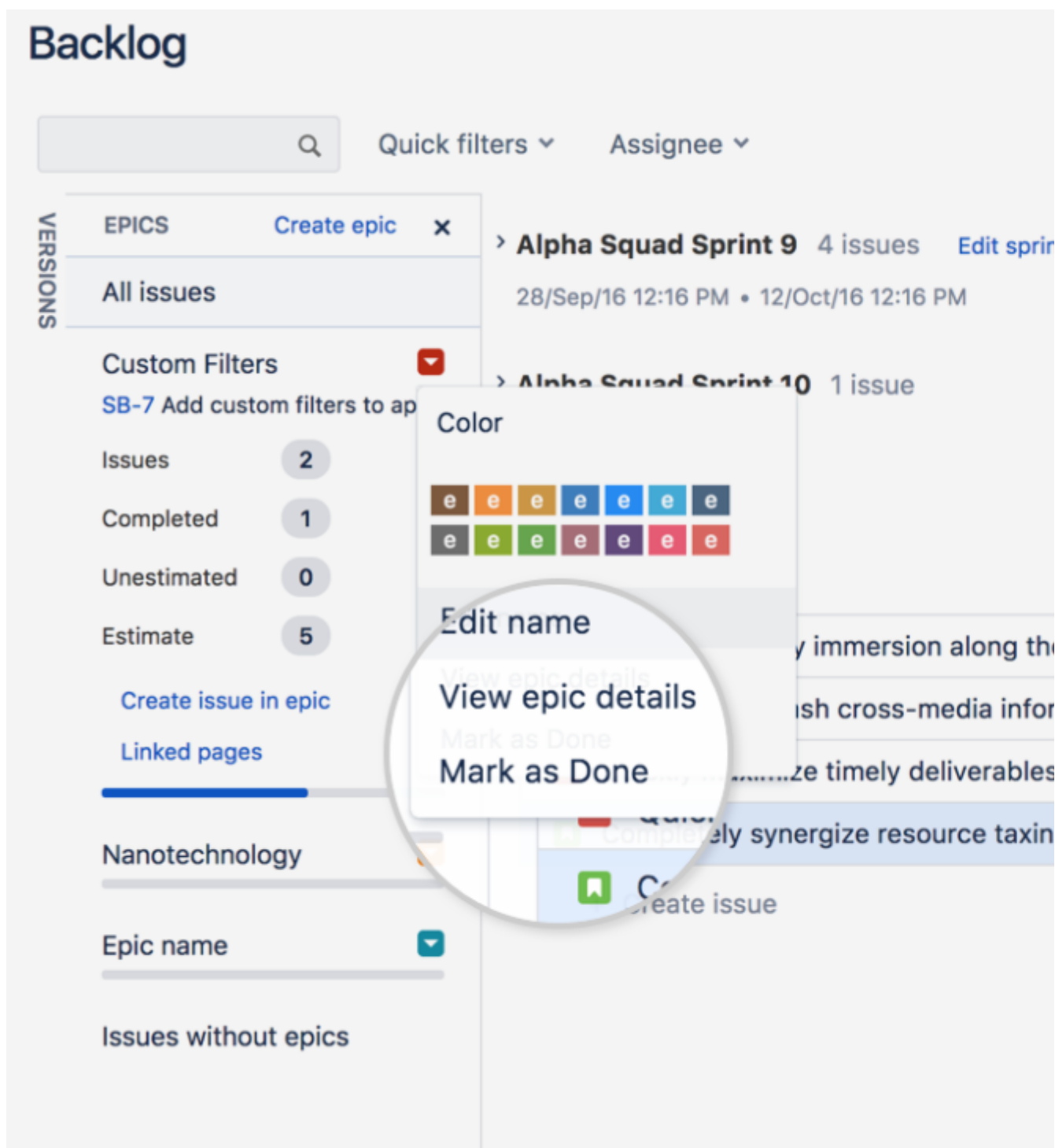
ⓘ Marking an epic as Done doesn't have any impact on the workflow status of the Epic or its linked issues.

Re-opening a Done epic

If you have marked an epic as Done, and you want to change this status, you need to do so using the Issue detail view.

The quickest way to update one or multiple epics is to do it using the bulk edit.

1. Do a JQL search of the epics to update.
2. Perform a **Bulk Change > Edit Issues.**
3. In the **Change Epic Status** field, select **To do** or **In Progress**.

☐ Change Labels

Add to existing ▾

▾

Begin typing to find and create labels or press down to select a suggested label.

☐ Change Time Tracking   ☐ Change Original Estimate   [   ] (eg. 3w 4d 12h) ⑦

The original estimate of how much work is involved in resolving this issue.

☐ Change Remaining Estimate   [   ] (eg. 3w 4d 12h) ⑦

An estimate of how much work remains until this issue will be resolved.

☐ Change Development

☐ Change Epic Colour

ghx-label-1

Epic Colour field for Jira Software use only.

☐ Change Epic Link

An epic cannot have another epic linked to it.
Choose an epic to assign this issue to.

☐ Change Epic Name

Provide a short name to identify this epic.

☐ Change Epic Status

To Do ▾

Epic Status field for Jira Software use only.

☐ Change Original story points

☐ Change Parent Link

An issue at the top of the Portfolio for Jira hierarchy cannot have a parent set.
Choose a parent to assign this issue to.

Workflow statuses

The path that your issues take is called a workflow. Each Jira workflow is composed of a set of statuses and transitions that your issue moves through during its lifecycle, and typically represents work processes within your organization. Workflows can be associated with particular projects and, optionally, specific issue types by using a workflow scheme.

The status field you see on the epic page is the issue status (epic being a type of an issue) as specified in the workflow, not the epic status.

JIRA IX / JIX-455

Content Design work for J

✎ Edit   **🔍 Comment**   **Assign**   **More ⌄**

⌄ Details

Type:	⚡ Epic
Priority:	⌃ Highest
Affects Version/s:	None
Component/s:	None
Labels:	None
Epic Name:	Jira 8.0

Align workflow status with epic status

To do that, you need to create a new workflow and associate it with the Epic Issue Type.

Before you begin

Get familiar with which Workflow Statuses map to which Epic Statuses, and make sure you have them ready. You'll also need the Jira Suite Utilities plugin. We also recommend backing up your data just in case.

- Workflow Status describes how to manage workflow statuses.
- Configuring a Custom Field - as Epic Status is a custom field, it is configured using these instructions.
- JIRA Suite Utilities is available through the Marketplace.

Copy your Existing Workflow

1. Go to **Jira Administration > Issues > Workflows**.
2. Copy the Workflow associated with your existing project.
3. Rename the Workflow.

Add the Post Functions

1. Select **Edit** to update the workflow.
2. Select all transitions one by one .
3. Select **Post Functions**.
4. Select **Add post function**.
5. Select **Update Issue Custom Field** and select **Add**.
6. In the Issue Custom Field drop-down select Epic Status.
7. Add the **Custom Field Value** that matches the Status which the transition is associated with. For example, if you are adding the transition for To Do then enter To Do; for Done, enter Done.)
8. Select **Add**.
9. Repeat this task for all the statuses in your workflow.

Associate the Workflow with the Epic Issue Type

1. Go to **Administration > Issues > Workflow Schemes.**
2. Click **Edit** for the workflow associated with your project.
3. Select **Add Workflow > Add Existing** and then choose your new workflow.
4. Select **Epic Issue Type.**
5. **Publish** your new Workflow.

Now, when you change the status of the Issue, the Epic Status will reflect the Issue Workflow status.

Getting to work

Ready to start working? You should have built a backlog and planned your version and epics by this stage. Time to get your team working!

The documentation in this section will help you use Jira Software to coordinate your team's work. For Scrum, these tasks include planning and running sprints. For Kanban, these tasks include monitoring work in progress. Both Scrum and Kanban use boards (Active sprints/Kanban board) and wallboards to keep your team on track.

Search the topics in 'Getting to work':

Overview

Working in a Scrum project	Scrum projects are based around working in sprints. A sprint is an iteration of work (two to four weeks) that is typically part of a version. Each sprint starts with a planning meeting where the work is estimated and assigned to the sprint. At the end of each sprint, a retrospective or review meeting is held before the next sprint is planned. In Jira Software, sprints are planned using the **Backlog** of a board. Once a sprint has started, the sprint is monitored using the **Active sprints** of a board. Learn more: Running sprints in a Scrum project, Checking the progress of a version, and Using Active sprints
Working in Kanban project	Unlike Scrum, Kanban doesn't use the concept of sprints. Rather, a Kanban project is based on the continuous delivery of work. When tasks are completed, more tasks are pulled into the work-in-progress pipeline. Constraints are placed on the amount of work that can be done at one time, so that the team is not overburdened. In Jira Software, work is managed and monitored using the **Kanban board**. Learn more: Monitoring work in a Kanban project

Already finished work for the version?

Next: Release your version

Project lifecycle					
NEW PROJECT	BACKLOG	NEW VERSION	WORK	RELEASE	REPORTING

Running sprints in a Scrum project

A *sprint* — also known as an *iteration* — is a short period in which the development team implements and delivers a discrete and potentially shippable application increment, e.g. a working milestone version. If you haven't run sprints before, we recommend using a fixed two-week duration for each sprint. It's long enough to get something accomplished, but not so long that the team isn't getting regular feedback.

Note, sprints do not apply to Kanban projects.

In Jira Software, you view sprints on a board and assign issues to sprints. You can search for issues in upcoming sprints using the **Sprint** field. You can view the **Sprint** field on each individual issue as well, to see the sprint that an issue is part of.

To learn more about running sprints, read the following topics:

- To learn how to create and modify a sprint, add issues to a sprint, and start a sprint, see Planning sprints.
- To find out how you can use Active sprints, wallboards, reports, and more, to track a sprint that is in progress, see Monitoring the progress of a sprint.
- To learn how to close a sprint and complete related tasks, like reporting, see Completing a sprint.
- To learn how to reopen a sprint that's already closed, see Reopening a sprint.
- To see how you can link Confluence pages (e.g. 'Retrospective notes') to a sprint, see Linking a Confluence page to a sprint.

Known issues

This table lists known issues in the behavior of and in your work with sprints. Here, you'll also find possible solutions and temporary workarounds.

We are aware of these issues and have planned their resolution in future releases.

Issue	Solution
Closed and completed sprints get automatically removed from the **Sprint** issue field when you're moving issues to a new project.	This is a known issue tracked in the ticket ⬛ JSWSERVER-13333 - Moving Jira issues to a new project will automatically delete closed/completed sprints from the Sprints field CLOSED See the Knowledge Base article to know more about the issue

Planning sprints

Every sprint starts with a planning meeting. When planning a sprint, your team would typically commit to deliver a set of stories that are pulled from the top of the backlog. In Jira Software, this involves creating a sprint, assigning stories to the sprint, and starting the sprint. The instructions on this page will help you complete these activities.

On this page:

- Before you begin
- Creating a sprint
- Adding issues to a sprint
- Starting a sprint
- Editing, reordering, or deleting a sprint
- Viewing the issues in a sprint
- Next steps

Before you begin

- Sprints only apply to Scrum boards.
- You must have ranking enabled on your board to use sprints. See En abling ranking.
- In general, sprint actions require the Manage Sprints permission. This is permission that gives you full rights to manage sprints. However, there are more granular permissions that give users the right to edit the goal and name of a sprint (Edit sprints) and start and end a sprint (Start/Complete sprints).
  There are also some sprint actions (e.g. adding issues to sprints, removing issues from sprints) that require the Schedule Issues and Edit Issues permissions.
  See Permissions overview for more information.

Creating a sprint

You can create a sprint for your current iteration, or multiple future sprints if you want to plan several iterations ahead.

1. Go to the **Backlog** of your Scrum project.
2. Click the **Create sprint** button at the top of the backlog.
3. Add a start and an end date to your sprint (these can be future dates as well) or leave both date fields empty.
4. Click **Create sprint**.
5. Once the sprint is created, click **More** (⋯) **> Add sprint goal** to add a goal for this sprint.

✅ *Tip: Need to record your planning meeting notes somewhere? If you have Jira Software connected to Confluence, you can create a 'Meeting Notes' page via the **Linked pages** link. See Linking a Confluence page to a sprint for details.*

Adding issues to a sprint

In Scrum, scope creep during a sprint should be avoided. In addition, the team is supposed to deliver a working piece of software at the end of the sprint. This means that you need to know your team's capacity for work, as well as the amount of work they are committing to, when adding issues to a sprint.

Typically, your team would estimate issues before adding them to the sprint, so that you can see the total estimated work for the sprint (in the sprint footer). You can match this against your team's capacity for work by looking at past sprints. Tools like the Velocity Chart and Burndown Chart can help you with this. Don't worry if you don't have any historical data — you'll get a good idea of your team's velocity once they start completing sprints.

Note:

- An issue cannot belong to more than one sprint.
- A sprint can include issues from more than one project, if the board filter includes multiple projects.
- If you add an issue to a sprint on one board, that particular sprint will also appear on all other boards that contain that particular issue.
- Sub-tasks cannot be moved independently of their parents.

There are three ways to add an issue to a sprint:

Use this when the issue already exists, and you want to add the issue to an active or future sprint.

In the **Backlog**, drag and drop the issues onto the relevant sprint.

If you want to add multiple issues you can either:

- select the issue(s) (use Shift+Click or Ctrl+Click), right-click, then select the relevant sprint, or
- drag the sprint footer down to include issues from the backlog.
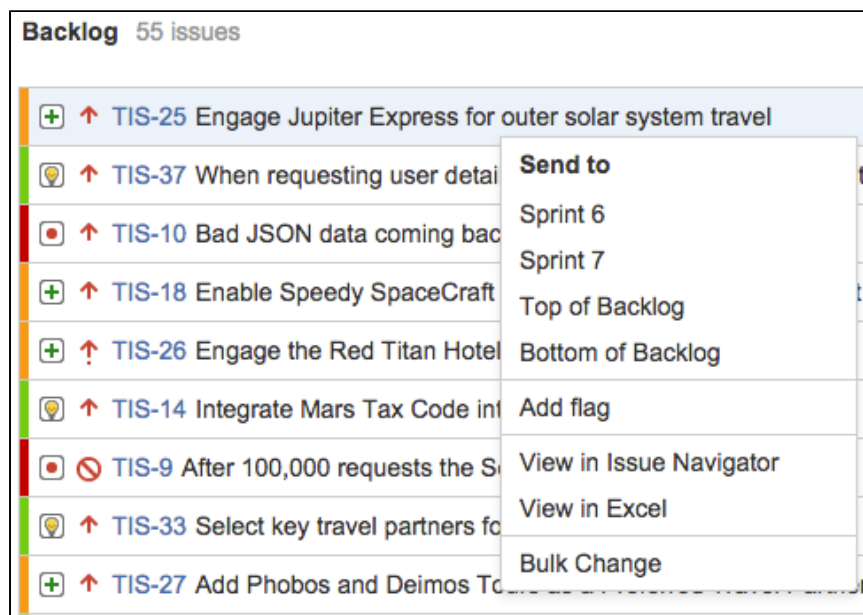
Use this when the issue doesn't exist, and you want to quickly add it to an active sprint.

After creating an issue in the **Active sprints**, click the **Add to <sprint name>** link in the confirmation dialog that displays. **<Sprint name>** will be the name of the sprint that you are currently viewing on the board.

Note, if you do not have the 'Edit Issues' and 'Schedule Issues' permissions for all projects included by the board's filter, the issue will be added to the backlog instead of the sprint.
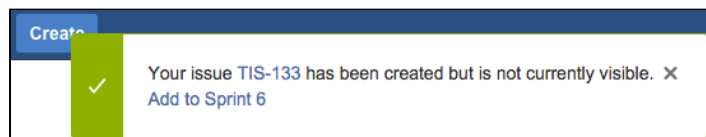
Use this when editing an issue, and you know the name of the sprint (active and future sprints only).

Create or edit an issue and enter the sprint name in the **Sprint** field. If the Sprint field doesn't display on the Create Issue or Edit Issue dialog, choose **Configure Fields**, then select the **Sprint** field.

Note, if you update the Sprint field for multiple issues via bulk operations, you need to enter the Sprint ID, not Sprint name. To find the Sprint ID, navigate to an issue in the sprint, hover over the sprint name, and look in the URL for number in the **sprint** *parameter.*

If you want to remove an issue from a sprint, just drag it to the **Backlog**, or right-click and select **Send to Top of Backlog/Bottom of Backlog**. You can also edit or bulk edit issues to remove the value from the Sprint field of the issues.

Once you have chosen which issues to include in your next sprint, you are ready to start the sprint, as described in the 'Starting a sprint' section below.

Starting a sprint

You can only start a sprint, if:

- You haven't started one already. If you want to have more than one active sprint at a time, try the Para llel Sprints feature.
- The sprint is at the top of the backlog. If you want to start a planned sprint that is lower down, you will need to reorder your sprints to move it to the top. (Note, this doesn't apply if Parallel Sprints is enabled).

To start a sprint:

1. Go to the **Backlog** of your Scrum project.
2. Find the sprint that you want to start and click **Start sprint.**
   If you haven't estimated your issues, you will receive a warning when starting the sprint (unless you are using Issue Count, as this is calculated automatically).
3. Update the **Sprint Name**, if desired, and, if you have not done it already, specify the **Start Date** and **E nd Date** for the sprint.
   Note, the default duration of a sprint is two weeks.

You will be taken to the **Active sprints**, where issues in your newly started sprint will be shown.

You can still view your active sprint in the **Backlog**, and even add and remove issues from it. Be aware that adding and removing an issue from the active sprint is considered to be a 'scope change' and will be reflected in some reports, e.g. the Burndown Chart.

Editing, reordering, or deleting a sprint

Sometimes, you may need to modify sprints after they have been created, particularly if you are managing a large backlog. You can edit the name and dates of a sprint, reorder a sprint, or delete a sprint in the Backlog.

To edit the name or dates for a sprint, click **More** (···) > **Edit sprint**. Note, you can only edit sprints that have not been completed yet.

To delete a sprint, click the ··· icon (next to **Start sprint**) for the sprint that you want to move, then click **D elete sprint**. Note, you cannot delete the active sprint, you can only complete it. You also cannot delete closed (i.e. completed) sprints.

To reorder a sprint, click the ··· icon (next to **Start sprint**) for the sprint that you want to move, then choose **Move sprint up** or **Move sprint down.** Note, you can only reorder planned sprints (not the active sprint or closed sprints). Also, if you have the same sprints (or some subset of them) on different boards, the sprint order will be reflected across the boards.

Viewing the issues in a sprint

Use the Backlog of a board to view planned sprints. If you want to view a sprint in progress, use the Active sprints of a board instead. You can also use JQL in the issue search (**Issues** > **Search for issues** > **Advanc ed**) to search for a sprint's issues.

For example, to search for issues that are in the sprint, "February 1", use this query:

```
sprint = "February 1"
```

For more information on how to search for issues by sprint:

- See the JQL documentation on the `sprint` field ('Fields Reference' section),
- See Advanced searching functions on the `closedSprints()` and `openSprints()` functions.

Next steps

> (i) **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Using your Scrum backlog
- Estimating an issue
- Burndown Chart
- Monitoring the progress of a sprint
- Completing a sprint

Monitoring the progress of a sprint

You've started a sprint with grand plans. How do you make sure that your team is on track to deliver what they've committed to?

Jira Software provides a number of tools that you can use to monitor the progress of a sprint. If you run into problems or things are just not on track for the target date, Jira Software can help visualize or highlight problems so your team can take actions to remedy them.

On this page:

- Active sprints
- Development panel for issues
- Wallboards and gadgets
- Burndown chart
- View the Burnup Chart

Active sprints

The Active sprints of a scrum board just may be the central workspace for a sprint that is in progress. You and your team would usually check Active sprints at least once a day, while a sprint is running — perhaps your team has its daily stand-ups around a monitor displaying Active sprints.

Learn more: Using Active sprints
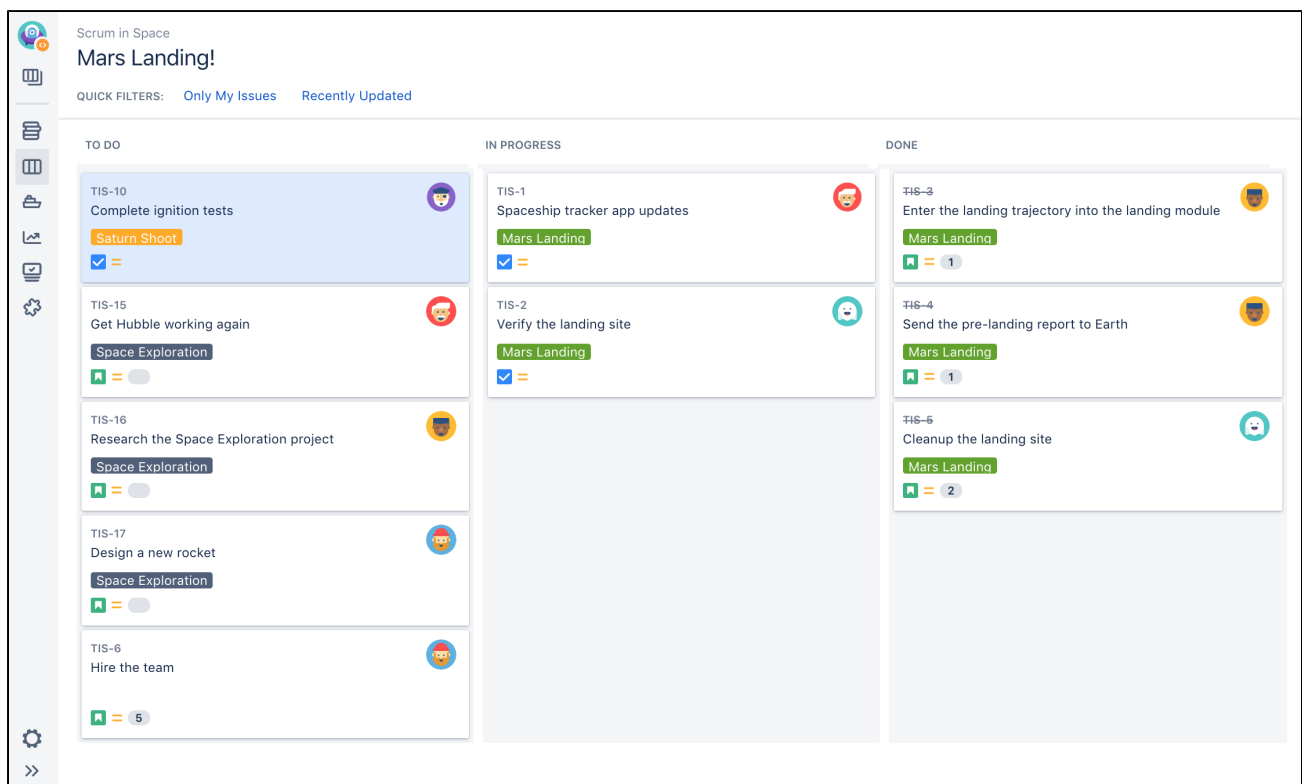
Development panel for issues

If you have Jira Software connected to your development tools, a 'Development' panel is shown on each issue that shows you information about the commits, branches, pull requests, builds, and more, that are related to the issue. This can help you see discrepancies at a glance — for example, issues that have related commits, but are still in the 'To Do' status.

Learn more: Configuring development tools
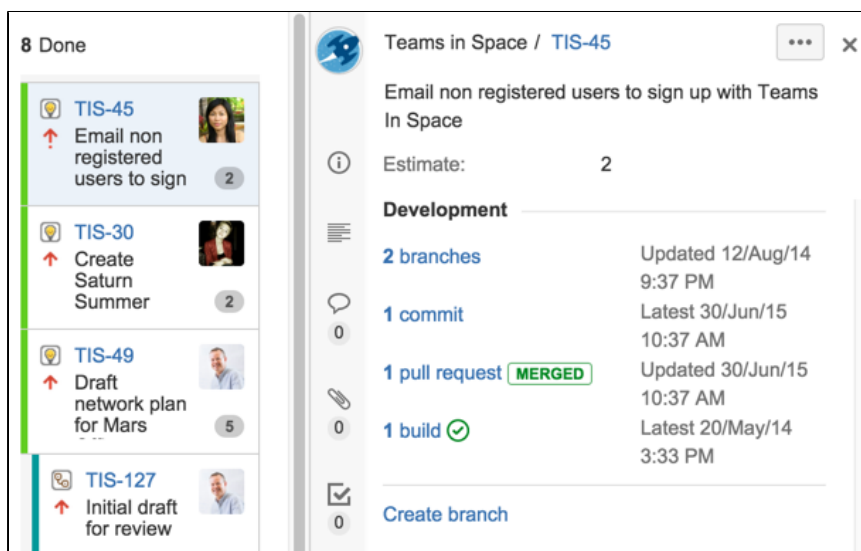
Wallboards and gadgets

A wallboard can act as an "information radiator" for your team, letting them know how they are progressing during a sprint, alerting them when builds are broken, reminding them about upcoming team events, and more. You just need to add the desired gadgets to a dashboard, then display it as a wallboard (**Tools** > **View as Wallboard** on the dashboard).

Learn more: Configuring dashboards
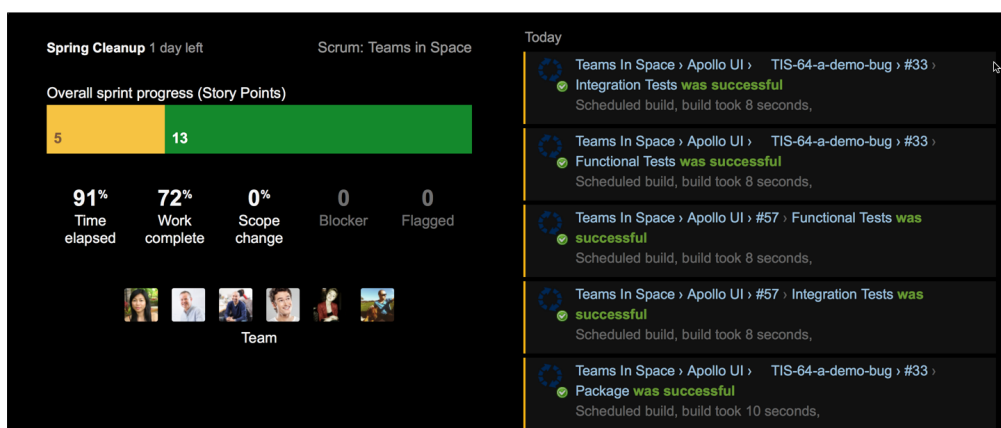
Burndown chart

The Burndown chart shows the work that has been completed in a sprint and estimates how long it will take to complete the remaining work. This can help you determine the likelihood of your team completing a sprint on time — allowing you to visualize or highlight problems so your team can take actions to remedy them.

Learn more: Burndown Chart

View the Burnup Chart

The Burnup Chart provides a visual representation of a sprint's completed work compared with its total scope. It offers insights on your project's progress, as well as offers warnings to help you maintain your project's health; you can instantly identify problems such as scope creep or a deviation from the planned project path.

Burn-up charts has one important advantage over the Burndown Chart in that it allows to divide the scope and the progress. Both scope and progress are combined on the Burndown chart and it is not possible to visualize and identify changes in scope and/or progress.

Learn more: Burnup Chart

Burnup Chart Switch report ⌄

Board ⌄

1925 ⌄   Story Points ⌄   ⓘ How to read this chart

／ Work scope   ⁄⁄ Scope projection   ／ Completed Work   ／ Guideline

Today

9/11/19 1:37 PM
Work scope   23
Completed Work   8

STORY POINTS

20

15

10

5

0

Nov 4   Nov 5   Nov 6   Nov 7   Nov 8   Nov 9   Nov 10   Nov 11   Nov 12   Nov 13

TIME

Completing a sprint

The end of a sprint is usually the time where your team takes stock of its progress. This usually includes demonstrations of the work completed during the sprint, followed by a sprint retrospective to analyze where improvements can be made.

As the team lead, Scrum master, or product owner, you can also use this time to check how your team is progressing against the overall version, and provide feedback to your stakeholders.

Before you begin

- Sprints only apply to Scrum boards. If you are using a Kanban board, you may want to read this instead: Deploying a release.
- To complete a sprint, you must be a Jira Administrator, or a user with the Manage Sprints permission, or Start/Complete sprints permission.

If you can't complete a sprint, your board's filter may be complex, and Jira Software is unable to determine which projects will be returned by the query. See Using Manage Sprints permission for advanced cases if you need help.

Completing the active sprint

1. Navigate to the **Active sprints** of your desired board.
2. If necessary, select the sprint you want to complete from the sprint drop-down.
3. Click **Complete sprint**.

If you have multiple sprints in the Active sprints of your board, the **Complete sprint** button will not appear until you select one of the multiple sprints.

> ⓘ If you're deleting a scrum board that contains an active sprint, you'll be asked to complete the sprint before deleting the board. So, you'll save the teamwork done during the sprint and won't have to restore the data.

When you complete the sprint, the following actions will occur:

- Your completed issues will move out of Active sprints.
- You can move any issues not completed at the end of the sprint to one of the inactive sprints, the Backlog, or a new sprint.
- For sprints that have parent issues and sub-tasks:
    - If you have parent issues that are 'Done' but sub-tasks that are not 'Done', you won't be able to end the sprint. You must complete the sub-tasks first.
    - If you have parent issues that are not 'Done' but have sub-tasks that are all 'Done', the parent issues will still be moved to the selected future sprint or to the Backlog. If these parent issues are part of another active sprint, the previously completed sub-tasks are still 'Done'.
- Your issues won't be marked with the date the sprint was closed; however, you can always view the sprint for an issue to find out when the sprint ended.

> ⚠ Closed and completed sprints get automatically removed from the **Sprint** issue field when you're moving issues to a new project. Learn more about the issue in the Knowledge Base article

Other tasks to complete at the end of a sprint

Conduct a retrospective and link your notes to the sprint

Typically, at the end of a sprint, you conduct a sprint retrospective. This is a meeting where your team discusses the completed sprint, including the things they need to start doing, stop doing, or continue doing. The outcome of this meeting should be actions for improvements in future sprints.

When you complete a sprint in Jira Software, you will be shown the 'Sprint Report', which provides helpful data about your sprint for your team's retrospective. If you have Confluence linked to Jira Software, you can also create your retrospective notes as Confluence pages and link them to your Sprint Report. See Linking a Confluence page to a sprint for more information.

Check the progress of your version

The end of a sprint is a great time to check how your version is progressing. See Checking the progress of a version for more information.

Release the sprint as a version (optional)

Your team may not need to release a version at the end of a sprint — but if you need to, it's easy to do. In the **Completed Issues** section of the Sprint Report, just click **View in Issue Navigator** . You should have already pre-assigned issues to a version before starting work on them, (which results in better reports). However, if you need to specify a version for issues at this stage, use Jira's **Bulk Edit** to assign all of the issues to the relevant version (see Editing multiple issues at the same time for more information). Note that you will not be able to do this if your 'Done' column sets an issue's status to 'Closed', as issues are not editable once they are 'Closed' (but 'Resolved' is fine). Also, note you can only **Bulk Edit** the 'Fix Version' for issues from one project at a time.

Start planning your next sprint!

Once you have completed a sprint in Jira Software, you can start getting ready for a new one. You can also plan a few sprints ahead adding start and end dates to your future sprints.

Note, you can only have one active sprint at a time, unless you've enabled Parallel Sprints.

Next steps

> (i) **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Sprint Report
- Planning sprints
- Reopening a sprint

Reopening a sprint

Jira Software lets you reopen sprints that have been previously closed. This is a handy feature when:

- You've mistakenly completed the wrong sprint.
- You completed the correct sprint, but you completed it too early.

Moreover, sprints that are mistakenly closed or closed too early will affect your team's velocity, as the data in these sprints will still be included in your board. Your Velocity Chart will then be reflecting inaccurate data.

> ⓘ Consider that sprints only apply to Scrum boards.

General points when reopening sprints

> ⓘ To reopen a sprint, you must be logged in as a user with the **Jira administrator** permissions or a user with the **manage sprints** permission. For details, see Permissions overview.

In general, it's worthy to keep these points in mind when you're trying to reopen a sprint:

- When the Parallel Sprints feature is not enabled and there is a currently active sprint, you cannot reopen the sprint you previously completed.
- When issues in the completed sprint have already been moved to other active sprints, you can still reopen the sprint, but it will not contain the issues that are already in the active sprints.
- For issues that are completed outside of a sprint, but are then pulled back into a sprint you just reopened, the Sprint Report for the opened sprint will show these issues as "completed outside of this sprint".

The following diagram shows you the possible flows when you're reopening sprints, depending on certain scenarios.

Questions are asked for every individual issue in Sprint 1

Reopen Sprint 1

[Issue completed?] — No → [Issue been completed in Sprint 1?] — No → [Issue completed before added to Sprint 1?]

- Reopen Sprint 1 — No → Issue completed?
  - Yes: Message displays: "Issue is in an active sprint" → Sprint 1 will be removed from the issue's historic sprints → Sprint 1 Report: "Issue Removed from Sprint"
- Issue completed? — No: Issue is moved back to Sprint 1 → Sprint 1 Report: "Issue Not Completed"
- Issue completed? — Yes → Issue been completed in Sprint 1?
  - Yes: Issue is moved back to Sprint 1 → Sprint 1 Report: "Completed Issues"
- Issue been completed in Sprint 1? — No → Issue completed before added to Sprint 1?
  - Yes: Issue is moved back to Sprint 1 → Sprint 1 Report: "Completed Issues"
  - No (Yes): Issue is moved back to Sprint 1 → Sprint 1 Report: "Completed Issues Outside of the Sprint"

Scenarios where you can reopen sprints

In the sample scenarios listed below, let's assume the following details:

- You've created a sprint and named it "Sprint 1".
- You added some issues to Sprint 1, and then started it.
- Your team is currently working on the issue in Sprint 1.
- You have a future planned sprint named "Sprint 2".

Simple scenarios

The following are some examples of simple scenarios where you may want to reopen a sprint, and the results you can expect.

Scenario details	Expected results

• You closed Sprint 1, and there were some completed issues in it. • All incomplete issues were moved back to the Backlog. • All these incomplete issues are still in the Backlog when you try to reopen Sprint 1.	1. Sprint 1 is reopened successfully. 2. Sprint 1 contains all the completed and incomplete issues that were in Sprint 1 when it was closed.
• You closed Sprint 1, and there were no completed issues in it. • All incomplete issues were moved back to the Backlog. • During sprint planning, you moved all of these issues to the future sprint, Sprint 2. • You started Sprint 2, and Sprint 2 is a currently active sprint when you try to reopen Sprint 1. • The Parallel Sprints feature is enabled.	Sprint 1 cannot be reopened because all incomplete issues in Sprint 1 when it was closed are now in the currently active Sprint 2.

Complex scenarios

In some cases, you may be able to reopen a sprint, but the results may not be as what you expect them to be. Or you may not be able to reopen a sprint altogether. The following are some examples of complex scenarios where you may want to reopen a sprint, and the results you can expect.

Scenario details	Expected results
• You closed Sprint 1, and there were some completed issues in it. • All incomplete issues were moved back to the Backlog. • During sprint planning, you moved some of these issues to the future sprint, Sprint 2. • You started Sprint 2, and Sprint 2 is a currently active sprint when you try to reopen Sprint 1. • The Parallel Sprints feature is enabled.	1. Sprint 1 is reopened successfully. 2. Sprint 1 contains all the completed and incomplete issues that were in the Sprint 1 when it was closed. 3. The incomplete issues that were in Sprint 1 when it was closed, but are currently in the active Sprint 2 will remain in Sprint 2. This is because issues in an active sprint are not moved during the reopen sprint process.

• You closed Sprint 1, and there were some completed issues in it. • All incomplete issues were moved back to the Backlog. • During sprint planning, you moved some of these issues to the future sprint, Sprint 2. • You started Sprint 2, and Sprint 2 is a currently active sprint when you try to reopen Sprint 1. • The Parallel Sprints feature is not enabled.	Sprint 1 cannot be reopened for the following reasons: • There is a currently active sprint, Sprint 2. • The Parallel Sprints feature is not enabled.
• You closed Sprint 1, and there were no completed issues in it. • All incomplete issues were moved back to the Backlog. • During sprint planning, you moved all of these issues to the future sprint, Sprint 2. • You started Sprint 2, and Sprint 2 is a currently active sprint when you try to reopen Sprint 1. • The Parallel Sprints feature is enabled.	Sprint 1 cannot be reopened because all of its issues are now in the currently active sprint, Sprint 2.

⚠ Closed and completed sprints get automatically removed from the **Sprint** issue field when you're moving issues to a new project. Learn more about the issue in the Knowledge Base article

Reopening a sprint

1. Navigate to your desired board.
2. Select **Reports**, and then select **Sprint Report**.
3. Select the relevant sprint from the sprint drop-down.
4. Select **Reopen Sprint**. The Reopen Sprint dialog will be displayed, showing you the outcome you can expect when the sprint is reopened.
5. Select **Reopen**.

Note the following details that take place when a sprint is completed then reopened:

• When you complete a sprint, this event of completing the sprint will be recorded in the corresponding Sprint Report.
• When you reopen that sprint, this event of reopening the sprint will also be recorded in the Sprint Report. However, the report will now use the new end date moving forward.
• When the reopened sprint is completed, the last completion date will appear as the end date in the Sprint Report.

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

DRAFT - Reopening a sprint

Jira Software lets you reopen sprints that have been previously closed. This is a handy feature when:

- You've mistakenly completed the wrong sprint.
- You completed the correct sprint, but you completed it too early.

Moreover, sprints that are mistakenly closed or closed too early will affect your team's velocity, as the data in these sprints will still be included in your board. Your Velocity Chart will then be reflecting inaccurate data.

Before you begin

- Sprints only apply to Scrum boards.
- To reopen a sprint, you must be a Jira Administrator or a user with the Manage Sprints permission.

General points when reopening sprints

In general, it's worthy to keep these points in mind when you're trying to reopen a sprint:

- When the Parallel Sprints feature is not enabled and there is a currently active sprint, you cannot reopen the sprint you previously completed.
- When issues in the completed sprint have already been moved to other active sprints, you can still reopen the sprint, but it will not contain the issues that are already in the active sprints.
- For issues that are completed outside of a sprint, but are then pulled back into a sprint you just reopened, the Sprint Report for the opened sprint will show these issues as "completed outside of this sprint".

The following diagram shows you the possible flows when you're reopening sprints, depending on certain scenarios.

Questions are asked for every individual issue in Sprint 1

```
                    Reopen Sprint 1

                        │
                        ▼
         ◇───No──→◇ Issue ───Yes──→◇ Issue been ───No──→◇ Issue ───No─
         │          completed?      completed in        completed before
        Yes            │            Sprint 1?            added to
         │            No               │                Sprint 1?
         │             │              Yes                   │
         │             │               │                  Yes        Yes─
         ▼             ▼               ▼                    │
  Message displays:  Issue is moved  Issue is moved ←───────┘
  "Issue is in an   back to Sprint 1  back to Sprint 1
   active sprint"
         │                                                 Issue is moved back
         ▼                                                   to Sprint 1
  Sprint 1 will be
  removed from the
  issue's historic sprints
         │
         ▼
  Sprint 1 Report:   Sprint 1 Report:  Sprint 1 Report:   Sprint 1 Report:
  "Issue Removed    "Issue Not         "Completed Issues" "Completed Issues
   from Sprint"      Completed"                            Outside of the Sprint"
```

Scenarios where you can reopen sprints

In the sample scenarios listed below, let's assume the following details:

- You've created a sprint and named it "Sprint 1".
- You added some issues to Sprint 1 and then started it.
- Your team is currently working on the issue in Sprint 1.
- You have a future planned sprint named "Sprint 2".

Simple scenarios

The following are some examples of simple scenarios where you may want to reopen a sprint, and the results you can expect.

Scenario details	Expected results

• You closed Sprint 1, and there were some completed issues in it. • All incomplete issues were moved back to the Backlog. • All these incomplete issues are still in the Backlog when you try to reopen Sprint 1.	1. Sprint 1 is reopened successfully. 2. Sprint 1 contains all the completed and incomplete issues that were in Sprint 1 when it was closed.
• You closed Sprint 1, and there were no completed issues in it. • All incomplete issues were moved back to the Backlog. • During sprint planning, you moved all of these issues to the future sprint, Sprint 2. • You started Sprint 2, and Sprint 2 is a currently active sprint when you try to reopen Sprint 1. • The Parallel Sprints feature is enabled.	Sprint 1 cannot be reopened because all incomplete issues in Sprint 1 when it was closed are now in the currently active Sprint 2.

Complex scenarios

In some cases, you may be able to reopen a sprint, but the results may not be as what you expect them to be. Or you may not be able to reopen a sprint altogether. The following are some examples of complex scenarios where you may want to reopen a sprint, and the results you can expect.

Scenario details	Expected results
• You closed Sprint 1, and there were some completed issues in it. • All incomplete issues were moved back to the Backlog. • During sprint planning, you moved some of these issues to the future sprint, Sprint 2. • You started Sprint 2, and Sprint 2 is a currently active sprint when you try to reopen Sprint 1. • The Parallel Sprints feature is enabled.	1. Sprint 1 is reopened successfully. 2. Sprint 1 contains all the completed and incomplete issues that were in Sprint 1 when it was closed. 3. The incomplete issues that were in Sprint 1 when it was closed but are currently in the active Sprint 2 will remain in Sprint 2. This is because issues in an active sprint are not moved during the reopen sprint process.

	Sprint 1 cannot be reopened for the following reasons:
- You closed Sprint 1, and there were some completed issues in it. - All incomplete issues were moved back to the Backlog. - During sprint planning, you moved some of these issues to the future sprint, Sprint 2. - You started Sprint 2, and Sprint 2 is a currently active sprint when you try to reopen Sprint 1. - The Parallel Sprints feature is not enabled.	- There is a currently active sprint, Sprint 2. - The Parallel Sprints feature is not enabled.
- You closed Sprint 1, and there were no completed issues in it. - All incomplete issues were moved back to the Backlog. - During sprint planning, you moved all of these issues to the future sprint, Sprint 2. - You started Sprint 2, and Sprint 2 is a currently active sprint when you try to reopen Sprint 1. - The Parallel Sprints feature is enabled.	Sprint 1 cannot be reopened because all of its issues are now in the currently active sprint, Sprint 2.

Reopening a sprint

1. Navigate to your desired board.
2. Select **Reports**, then select **Sprint Report** .
3. Select the relevant sprint from the sprint dropdown.
4. Select **Reopen Sprint**. The Reopen Sprint dialog will be displayed, showing you the outcome you can expect when the sprint is reopened.
5. Select **Reopen**.

Note the following details that take place when a sprint is completed and then reopened:

- When you complete a sprint, this event of completing the sprint will be recorded in the corresponding Sprint Report.
- When you reopen that sprint, this event of reopening the sprint will also be recorded in the Sprint Report. However, the report will now use the new end date moving forward.
- When the reopened sprint is completed, the last completion date will appear as the end date in the Sprint Report.

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Sprint Report
- Planning sprints
- Completing a sprint

Deleting a sprint

You might find it helpful to delete a sprint if:

- You've created a sample sprint to experiment with Software's features.
- You've accidentally started a sprint and don't want it to reflect in any reports.

Keeping these sprints can be annoying because they'll appear in menus and search results. They'll also affect reports like the Velocity Chart, compromising your team's future planning. Instead of keeping these sprints in your project, you can delete them.

Before you begin

- You can only delete inactive sprints.
- You need the **Manage Sprints** permission.

Deleting a completed sprint

To delete a completed sprint:

1. Navigate to your desired board.
2. Select **Reports**, then select **Sprint Report**.
3. Select the relevant sprint from the sprint drop-down.
4. Say your goodbyes, and select **More** (⋯) > **Delete sprint**.

When you delete a sprint, it's gone for good, and all uncompleted issues are moved to the next sprint in the list on a board. Issues that were completed during this sprint will remain.

Linking a Confluence page to a sprint

If you have linked your Jira instance to a Confluence instance, you can create and link Confluence pages to your sprints via Jira Software. For example, you may want to write up the sprint meeting notes in Confluence and link them to the sprint. You may also want to write your team's retrospective notes in Confluence and link them to the Sprint Report for the completed sprint.

Screenshot: linked pages for a sprint

Before you begin

- These instructions only apply to Scrum boards.

Creating a new linked Confluence page for a sprint

... in Active sprints:

1. Navigate to the **Active sprints** of your desired board.
2. Find the sprint that you want to create a linked page for. Click **Linked pages** at the top right of the sprint.
3. Click **Create page**. The 'Create' dialog in Confluence will be displayed, with the Meeting Notes blueprint selected (select a different blueprint, if you don't want to use it).

... from the Sprint Report:

1. Navigate to the Sprint Report, as described above.
2. Click **Linked pages** above the report chart.
3. Click **Create page**. The 'Create' dialog in Confluence will be displayed, with the Retrospectives blueprint selected (select a different blueprint, if you don't want to use it).

Linking an existing Confluence page to a sprint

... in Active sprints:

1. Navigate to the **Active sprints** of your desired board.
2. Find the sprint that you want to create a linked page for. Click **Linked pages** at the top right of the sprint.
3. Click **Link page**, search for the page you want, and select it.

... from the Sprint Report:

1. Navigate to the Sprint Report, as described above.
2. Click **Linked pages** above the report chart.
3. Click **Link page**, search for the page you want, and select it.

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Using Jira applications with Confluence
- Creating issues and sub-tasks
- Working with epics

Configuring a sprint to auto-start and auto-complete

You no longer need to go over multiple boards to start a new sprint or end a completed sprint. With the auto-managed sprints feature, you can schedule the automated start and end of current and future sprints on your instance.

To use the feature, you must have the **Manage sprints** project permission.

As a prerequisite for using auto-managed sprints, you must also enable the Parallel sprints option. It allows you to have multiple active sprints at the same time.

The feature flag for auto-managed sprints is enabled by default: `com.atlassian.jira.agile.darkfeature.sprint.auto.management.enabled`. But you can disable the feature if you want. Learn how to disable auto-managed sprints

To schedule the sprint auto-start and auto-complete:

1. Enable the Parallel sprints option.
2. Configure a sprint schedule when creating a new or editing an existing sprint.

On this page:

- Important things to know
- Enabling the Parallel sprints option
- Configuring a sprint auto-start and auto-complete
- How it works
- Disabling the feature

Important things to know

Auto-managed sprints have the following specifics:

- A sprint that doesn't have any issues can still start automatically.
- If a sprint has closed parent issues whose child issues aren't closed, the sprint won't complete automatically. For example, when some closed tasks have open sub-tasks, you'll need to close these sub-tasks first. After that, the sprint will auto-complete.
- When a sprint auto-completes, all the remaining open issues will be moved to the destination you select in the sprint configuration:
  - the backlog
  - a new sprint that Jira creates automatically
  - any existing future sprint
- If something has changed about the selected destination for the remaining open issues, Jira will show you a notification about this in the board view and the backlog view. You'll also find a relevant warning in the sprint settings.
  - Until the current sprint auto-completes, you can change the destination for the open issues in the sprint settings. Otherwise, the issues will move to the backlog by default.
- If you configured an auto-managed sprint, when it auto-starts or auto-completes, you'll get an email notification that Anonymous has made some changes to this sprint.

Enabling the Parallel sprints option

To be able to schedule an auto-start or auto-complete for a new sprint while another sprint is still in progress, you will need to enable **parallel sprints**.

You must be a Jira admin to enable the Parallel sprints option. If you aren't, contact your Jira admin about this.

To enable parallel sprints:

1. In the upper-right corner, select **Administration** > **Applications**.
2. In the left-side panel, select **Jira Software configuration**.
3. Select **Parallel sprints**.

Jira Software configuration

This page lets you enable and disable certain features of Jira Software.

Features ☑ **Parallel Sprints**

Allows a new sprint to be started when one is already in progress.

☐ **Sticky sprint headers**

Makes the sprint headers stick to the top of backlog so that the sprint information is visible on screen even when scrolling through large amount of issues

☐ **Days in column**

Shows dots representing the time an issue spent in a particular column on Jira board. Disabling improves performance.

☐ **Relevant sprints**

Limits sprint suggestions in the Sprint drop-down to sprints from the selected project when creating or editing an issue.

Configuring a sprint auto-start and auto-complete

You can schedule an auto-start and auto-complete for a sprint when:

- creating a future sprint
- creating a new sprint that you want to start now
- editing an existing sprint

To schedule an auto-start and auto-complete for a sprint:

1. Start creating or editing a sprint.
2. Select the dates when you want to start and end the sprint.
3. Select **Start and complete automatically**.

> ⚠ Make sure your Jira profile timezone matches your local timezone. If they differ, Jira will use the profile timezone to schedule auto-managed sprints. Learn how to change your profile timezone

4. In the drop-down list, select the destination where the remaining open issues will move after the current sprint auto-completes. You have the following options:
   a. **Backlog** – the open issues will move to the backlog after the sprint auto-completes. The backlog is also a default destination for the cases where Jira can't move open issues to a future sprint because it has been completed or deleted.
   b. **New sprint** – the open issues will move to a new sprint that Jira will create automatically. This sprint will have only a name. You can edit the sprint settings as needed.

      The name of an auto-created sprint will be the same as the name of the first sprint at the top of your backlog. For example, if the first sprint in the backlog is "April-May Sprint", the new auto-created sprint will have the same name – "April-May Sprint".

      If the name of the first sprint at the top of the backlog has a number, Jira will add 1 to this number and set the new number in the auto-created sprint name. For example, if the first sprint in the backlog is "Sprint 10", the new auto-created sprint will be "Sprint 11".
   c. **Any existing future sprint** – the open issues will move to an existing future sprint.

      This sprint can be completed, deleted, or started before your current sprint auto-completes. Jira will notify you about any changes with warning messages in the board view and in the backlog view.

      You'll be able to change the destination in the sprint settings before the sprint auto-completes. If you don't change the destination, the issues will move to the backlog by default.
5. Select **Save**.

Create sprint

This sprint contains **5 working days.** Learn more

Sprint Name:*	Sprint 10
Goal:	The MVP of a feature is ready.
Duration:	1 week ⌄
Start date:	29/Mar/23 11:38 AM 📅
End date:	5/Apr/23 11:38 AM 📅

☑ Start and complete sprint automatically

After sprint completes, move open issues to:*

Backlog ⌄

☐ Create another   **Create**   Cancel

The sprint will start and complete automatically on the dates you set. On the board, you'll see the number of days left until the sprint auto-complete. The sprint will also be marked with the [AUTO-MANAGED] label in the backlog view.

What can happen to a selected existing sprint and what you should do

In the sprint and backlog views, you'll see warnings when something changes about an existing future sprint you selected as the destination for moving open issues from the current auto-managed sprint.

In the following table, learn what can happen to the selected existing sprint and what you should do in each case.

Case	What Jira does by default	What you can do
The selected existing sprint has been completed before your current sprint is scheduled to auto-complete.	• Shows a warning in the sprint and backlog views as soon as the selected sprint has been completed • Moves the remaining open issues to the **back log** as soon as the sprint auto-completes	Go to the settings of the current sprint and select a different destination for the open issues
The selected existing sprint has been deleted before your current sprint is scheduled to auto-complete.	• Shows a warning in the sprint and backlog views as soon as the selected sprint has been deleted • Moves the remaining open issues to the **back log** as soon as the sprint auto-completes	Go to the settings of the current sprint and select a different destination for the open issues

The selected existing sprint has been started before your current sprint is scheduled to auto-complete.	• Shows a warning in the sprint and backlog views as soon as the selected sprint has been started • Moves the remaining open issues to the **sele cted sprint** as soon as the current one auto-completes ⓘ Although Jira can move the issues to the already running sprint, we don't recommend this. The moved issues will impact the team velocity, work scope, and sprint burndown.	Go to the settings of the current sprint and select a different destination for the open issues

How it works

Here are a few under-the-hood details explaining how the sprint auto-management works.

- Automation is written as a cron job that is scheduled as a Jira service. You can find the complete information on it in Scheduler administration. Search for `com.atlassian.greenhopper.service. sprint.services.SprintAutoStartStopJob`.
- The job is scheduled with an interval of 1 minute. The first run is scheduled right after Jira startup. You can change intervals by setting the following property: `sprint.auto.start.stop.run. interval.in.minutes`. Learn how to set the property
- The job runs once per cluster and may run on any node. This means you'll need to search logs on all nodes if there's a problem.
- You can find more details about errors, warnings, and the overall behavior of the feature in the file `atl assian-greenhopper.log` with the prefix `[SPRINT-AUTO-START-STOP]`. Learn more about log files in Jira

Disabling the feature

You can disable auto-managed sprints in two ways:

- Disable parallel sprints
- Manage the feature flags

Disabling parallel sprints

To disable the Parallel sprints option:

1. In the upper-right corner, select **Administration** > **Applications**.
2. In the left-side panel, select **Jira Software configuration**.
3. Uncheck **Parallel sprints**.

Managing the feature flags

To disable auto-managed sprints with the feature flags:

1. Disable the following dark feature flag: `com.atlassian.jira.agile.darkfeature.sprint. auto.management.enabled`. Make sure that no input with the wildcard `*sprint.auto. management.enabled` is present.
2. Add the following dark feature flag: `com.atlassian.jira.agile.darkfeature.sprint.auto. management.disabled`.

Checking the progress of a version

You've been monitoring the progress of your sprints and keeping your team on track for each iteration. How do know if you are on track to deliver the version?

Jira Software provides a number of tools that you can use to monitor the progress of a version. These tools can help you see problems early, as well as determine the likelihood of releasing a version on time.

Before you begin

- This page only applies to Scrum projects.

Release Hub

The Release Hub helps you understand the status of work for your version at a point in time. It shows a summary and a breakdown of the issues in the version, grouped by status. In addition, if you have connected Jira Software to your development tools, this page also warns you if your development tools are out of sync with Jira Software — for example, completed issues with open pull requests, completed issues with failing builds, completed issues that have unreviewed commits, etc.

1. Navigate to your project.
2. Click **Releases** > your desired version.

Learn more: Using the release page to check the progress of a version
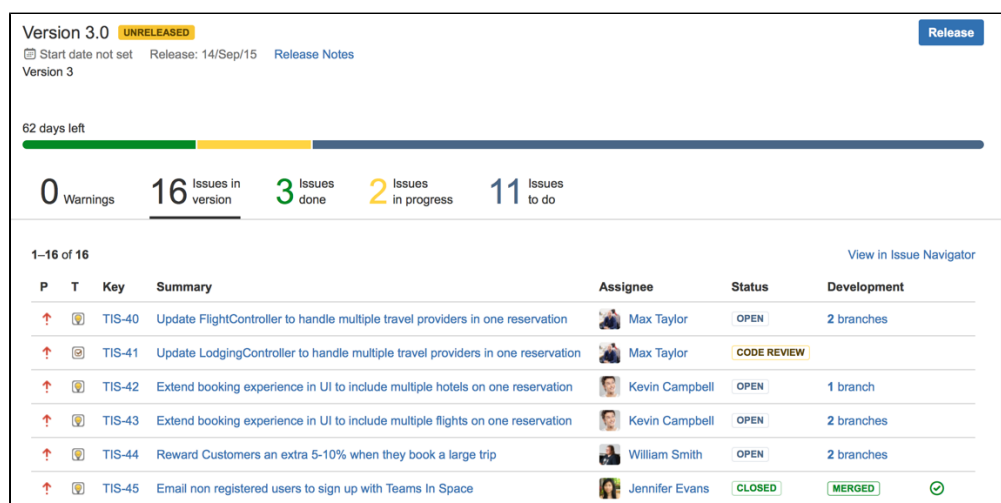
Release Burndown report

The 'Release Burndown' report helps you understand how your team is progressing towards the completion of the version. The report shows your team's velocity (relative to the current version), how scope changes have affected progress, and the estimated number of sprints needed to complete the remaining work.

Learn more: Release Burndown
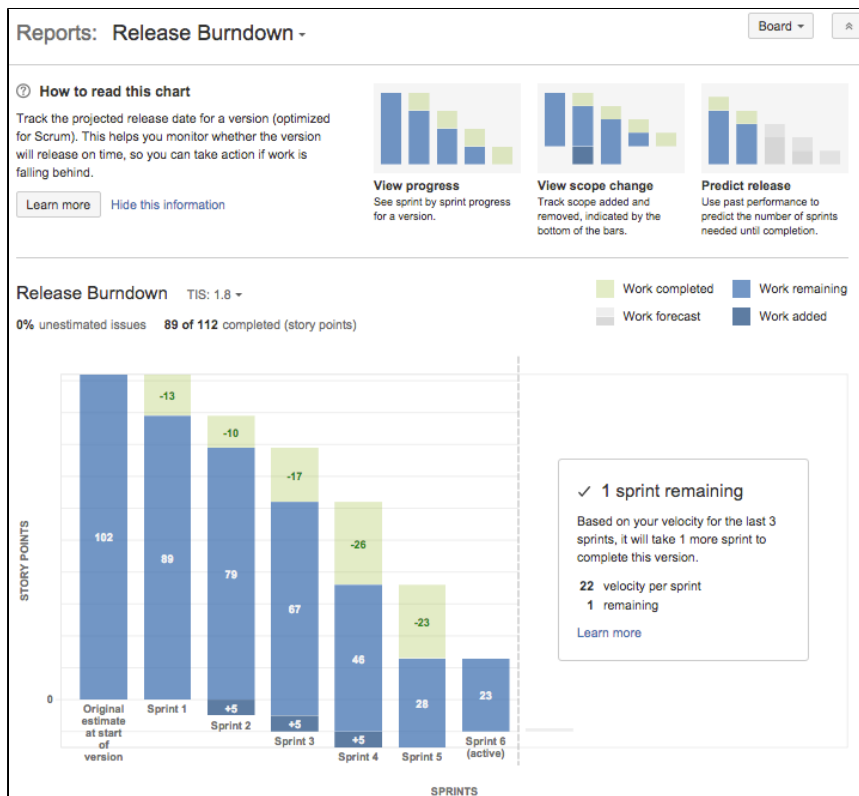
Status report

If you have Confluence linked to Jira Software and you need to provide status reports to external stakeholders, you can generate a 'Status Report' in Confluence that displays the progress of a version. See J ira Report Blueprint *(Confluence documentation).*

Using the release page to check the progress of a version
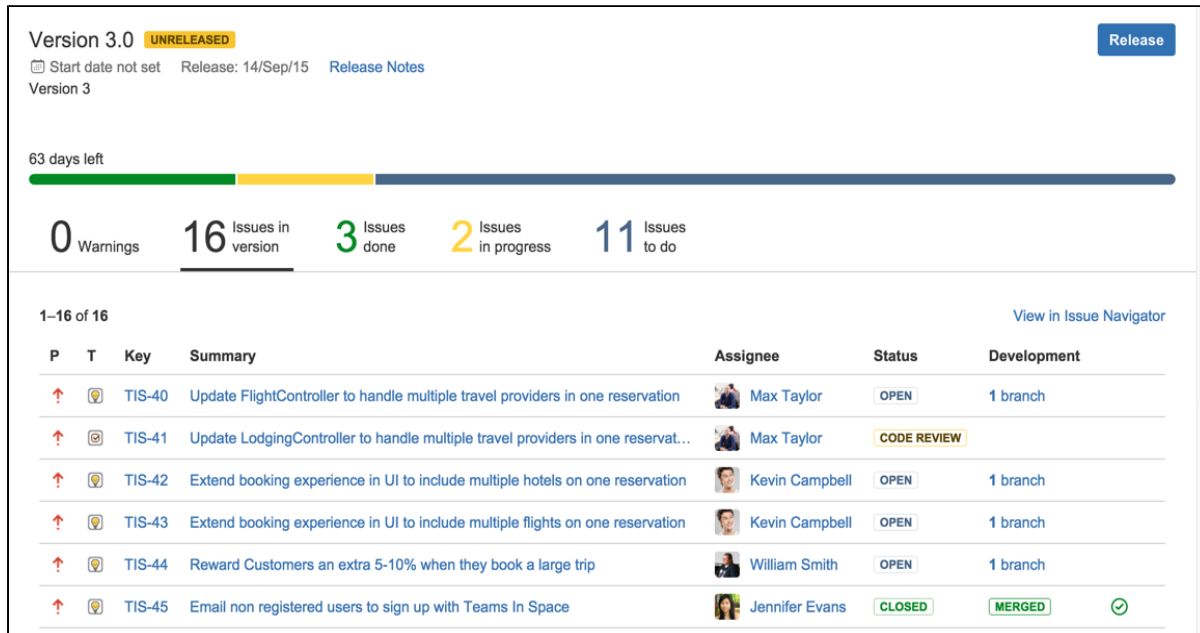
Screenshot: release page for a Jira version

Frequently asked questions

How do I find the 'Release' page for a version?

Navigate to your project > **Releases** > your desired version.

What warnings are shown on the 'Warnings' tab?

Issue warnings tell you when your issue data is potentially out of sync with your development data. Depending on which development tools are connected to Jira Software, the Release page will display the following warnings:

Warning type	Required development tool	When will a warning display?
Open pull requests	Bitbucket	An issue is done, but has an open pull request
Open reviews	Crucible	An issue is done, but has an open review
Unreviewed code	Crucible, Bitbucket	An issue is done, but its commits are not part of a pull request or code review
Failing builds	Bamboo	An issue is done, but is linked to a failing build

What statuses are considered to be 'To Do', 'In Progress', and 'Done'?

On the Release page for a version, issues are grouped into tabs, depending on whether they are 'To Do', 'In Progress', or 'Done'. Also, issue warnings are evaluated, depending on whether an issue is 'done' or not.

'To Do', 'In Progress', or 'Done' map to the status category of an issue's status. The status category of an status is defined when you create a status. See Defining status field values (Jira Admin documentation) for more information.

For example, the status 'Open' may have its status category set to 'To Do'. Therefore, it will display on the Release page in the 'Issues to do' tab.

Why can't I see any warnings?

Warnings will only display on the Release page if:

- Jira Software is connected to your Atlassian development tools, e.g. Bitbucket, Bamboo, Fisheye /Crucible; or GitHub. See Integrating with development tools for more information.
- You have the 'View Development Tools' permission. See Permissions overview for more information.
- Warnings are enabled. You can see which warnings are enabled by clicking the **Manage warnings** button.

What information will I see in the 'Development' column?

1. **Development panel**

The 'Development' column will display the commits, builds, and deployments related to an issue, provided that Jira Software is connected to the required development tool.

Information	Required development tool	Examples
Commits	Bitbucket, GitHub, Fisheye	3 commits
Branches	Bitbucket, GitHub	1 branch
Pull requests	Bitbucket, GitHub	2 pull requests OPEN

Builds	Bamboo	⊘ or ⚠
Deployments	Bamboo	✅

Click any of the symbols to view a detailed dialog. The dialog will also provide links to the commits, builds, or deployments in the relevant development tool.

How do I release a version?

Releasing the version will mark it as 'Released' in Jira Software. You will have the option of setting a release date and moving any incomplete issues to another version. If Jira Software is connected to Bamboo, you can also choose to run a build when the version is released. See Deploying a release to know more about releasing a version.

Why does a 'Syncing' message display next to my version's status?

Jira Software regularly synchronizes issues with their related development data. However, to ensure that you are viewing the most recent data when you visit the Release page for the version, Jira Software will perform another check and synchronize any data that is not up to date.

Monitoring work in a Kanban project

The Kanban board is a board that was created using the "Kanban" preset (see Creating a board).

Kanban is based on the continuous delivery of work. Rather than plan iterations, the flow of work is constantly monitored to ensure that there are always tasks being worked on. This means that when tasks are completed, new tasks are pulled into work-in-progress.

Use the Kanban board if your team focuses on managing and constraining work-in-progress.

- Before you begin
- About the Kanban board
- Accessing the Kanban board
- What can I do in a Kanban board?
- Next steps

Before you begin

You need to configure the columns for your board. When configuring the columns, make sure to:

- Assign to each state (column) a distinct function in your development process, e.g. in progress, testing, awaiting merge, etc.
- Configure appropriate constraints for the columns.

About the Kanban board

An issue will only be visible in the Kanban board if:

- the issue matches the board's saved filter,
- the issue's status maps to one of the board's columns (but not the 'Done' column), and
- there is at least a status being mapped to the right-most column, e.g. if you have the columns **Selected for Development, In Progress,** and **Done,** ensure that you have a status mapped to **In Progress** at least. If you map all the statuses to the first column (**Selected for Development**), you will not be able to see any issues in the Kanban board.

Accessing the Kanban board

Navigate to your desired board and select **Kanban board**.
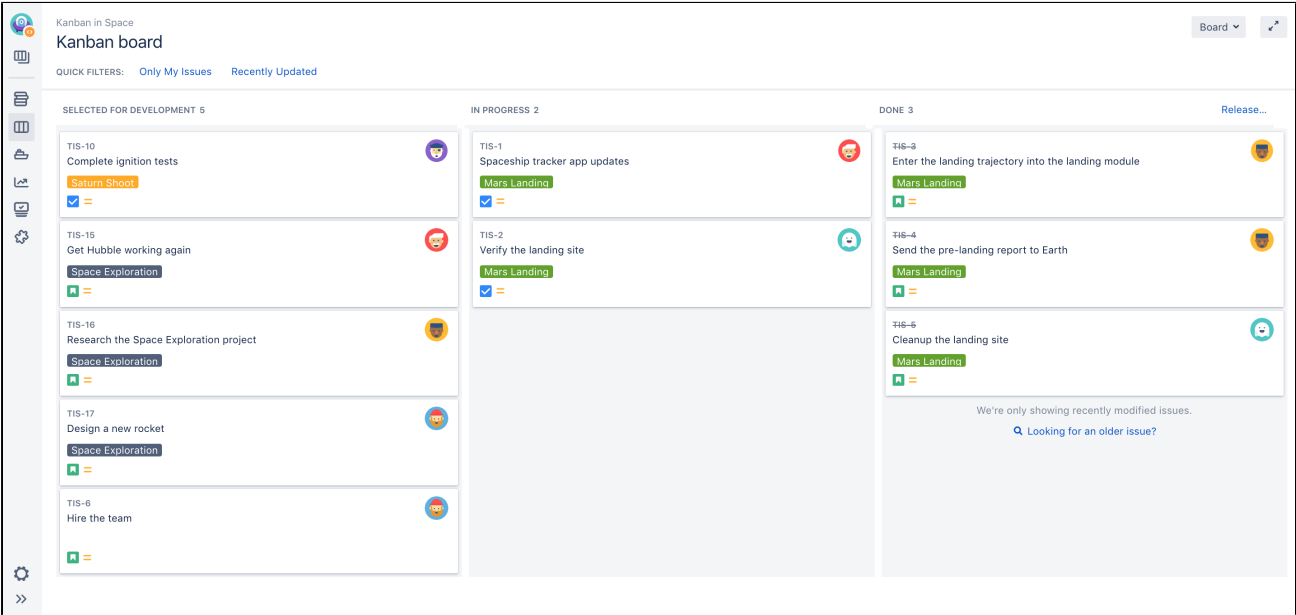
What can I do in a Kanban board?

Task	Instructions

179

Add issues to the Kanban board	Select **Create** in the header to open the 'Create Issue' dialog and create your issue. The issue will be added to the bottom of the Kanban board. Tip: tick the **Create another** checkbox in the 'Create Issue' dialog to keep it open, if you are creating multiple issues.
Prioritize the Kanban board	Drag and drop an issue to rank it. You can also right-click the issue to open a menu that allows you to send it to the top or the bottom of the column.
View an issue's details	Select the desired issue in the board. The issue's details will display in a panel on the right of the board. For information on how to configure the issue detail panel, see Configuring the issue view. If you want to select an issue rather than open the issue's details, use ctrl-left-click or command-left-click.
Estimate stories	Use the 'J' and 'K' keys to move through issues in the board and show the details on the right-hand side of the screen. Use the 'E' key to edit an issue, then update estimates or story points as you go. By default, the Story Points field is only available to issues of type 'Story' or 'Epic', see Configuring estimation and tracking for details.
Identify the workload for specialists	The avatars for users (specialists) who have work assigned to them in a sprint are shown at the top of a sprint. Select ... (next to avatars) to view the sprint workload for assignees.
Create sub-tasks	Select the desired issue, press the '.' key, and type 'create sub-task' to open the 'Create Sub-task' dialog. Create the sub-task as desired. Sub-tasks are useful for breaking a story (issue) down into implementable chunks.
Flag an issue	Right-click an issue to open a menu, and select **Add flag**.
Filter issues	Select the desired Quick Filter (below the board name), e.g. **Only my issues**. See Configuring Quick Filters for details.
Release issues	Select **Release** to release completed issues. When you release issues, these issues no longer appear in the **Done** column, and a release version is created, which you can view in the Releases tab of the project.
Hide issue detail view	If you're switching between the issues quite often, it might be useful to hide the issue detail view that appears on the right of the backlog. You can do it by selecting **More** `···` > **Hide detail view**.

Next steps

> ℹ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Ranking an issue
- Estimating an issue
- Working with epics

Using Active sprints

The Active sprints of a Scrum board displays the issues that your team is currently working on. You can create and update issues, and drag and drop issues to transition them through a workflow.

Before you begin

- Active sprints are only available on Scrum boards. If you are using a Kanban board, see Monitoring work in a Kanban project instead.
- Different functions on the Active sprints of a board require different permissions. For example, to remove an issue from a sprint, you need to have the Schedule Issues and Edit Issues permissions. See Permissions overview for more information.

Using Active sprints of a Scrum board

Navigate to the **Active sprints** of your desired board.
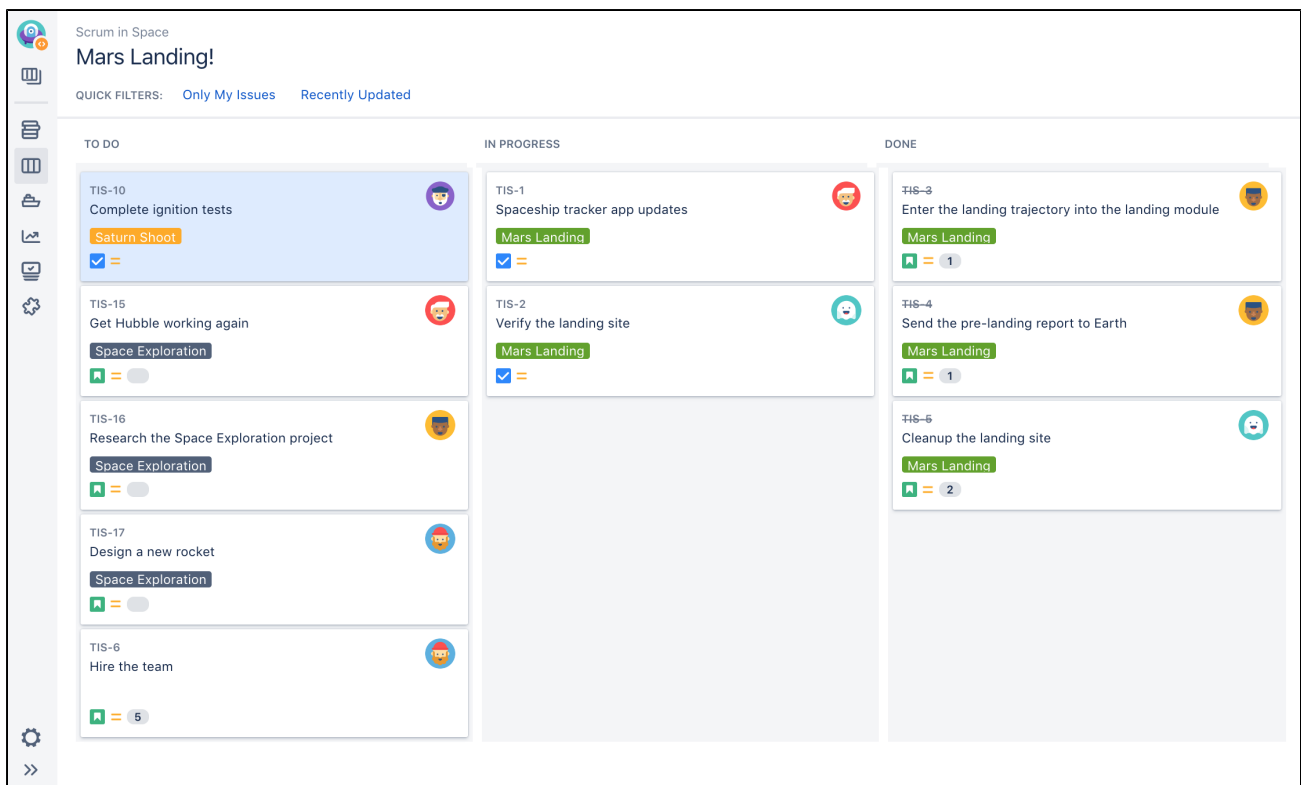
About Active sprints:

- An issue will only be visible in Active sprints if:
    - the issue's status maps to one of the board's columns,
    - the issue is in an active sprint (for Scrum boards), and
    - the issue matches the board's saved filter.
- Each column maps to one or more JIRA statuses.
- Sub-tasks are shown slightly indented, with their parent issue key shown above.
- Swimlanes are based on criteria of your choice. Issues hidden by a Quick Filter will be excluded from the swimlane count.
- The board will be refreshed every 30 seconds, if the browser window doesn't currently have focus. If you have focus on the window, it will show a message, asking you to refresh.
- If you've enabled the 'Days in column' indicator for your board (via configuring columns), dots will show at the bottom of each issue (up to the width of the card or a maximum of 32 dots). The dots indicate the number of days that the issue has been in its current column. Hover over the dots to see the total number of days.

What can I do in the Active sprints of a Scrum board?

Task	Instructions
Add issues to the 'To Do' column	Click **Create** in the header to open the 'Create Issue' dialog and create your issue. The issue will be added to the **To Do** column. Tip: tick the **Create another** checkbox in the 'Create Issue' dialog to keep it open, if you are creating multiple issues.
View an issue's details	Click the desired issue on the board. The issue's details will display in a panel on the right of the board. For information on how to use this issue detail panel, see Searching for Issues. You can also select multiple issues, right-click, then select **View in Issue Navigator** for a different view of the issues.
Prioritize an issue	Drag and drop an issue within a column to rank it.
Transition an issue	Drag and drop an issue between columns.
Create Sub-tasks	Select the desired issue, press the '.' key, and type 'create sub-task' to open the 'Create Sub-task' dialog. Create the sub-task as desired. Sub-tasks are useful for breaking a story (issue) down into implementable chunks.
Flag issues	Right-click (or select multiple issues and right-click) to open a menu and select **Add flag**.
Filter issues	Click the desired Quick Filter (below the board name), e.g. **Only My Issues**.
Bulk change issues	Right-click (or select multiple issues and right-click) to open a menu and select **Bulk Change**.
Remove issues from a sprint	Right-click (or select multiple issues and right-click) to open a menu and select **Remove from Sprint**.
Hide detail view	If you're switching between the issues quite often, it might be useful to hide the issue detail view that appears on the right of the backlog. You can do it by clicking **More** (···) **> Hide detail view**.

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read these related topics:

- Configuring columns
- Configuring filters
- Configuring Quick Filters
- Transitioning an issue
- Configuring swimlanes

Using Parallel Sprints

The Parallel Sprints feature lets you enable multiple active, parallel sprints. For example, if you have two teams working from the same backlog, each team can now work on their own sprint.

Please note the following caveats when using Parallel Sprints:

- The Velocity Chart will not show the velocity per team.
- The current implementation assumes that the teams perform estimation identically, which is unlikely in practice.

Enabling Parallel Sprints

1. Log in as a user with the 'Jira Administrators' global permission.
2. Select **Administration** (⚙) > **Applications**, then scroll down the page to the **Jira Software** section.
3. Under **Jira Software configuration**, select the **Parallel Sprints** checkbox.

Limiting sprint selection to relevant sprints

To speed up assigning issues to sprints, you can limit the sprint selection in the Sprint drop-down on the Create an issue and Edit an issue page to show only the sprints from the project you first select on these pages.

Enabling Relevant sprints

1. Log in as a user with the 'Jira Administrators' global permission.
2. Select **Administration** (⚙) > **Applications**, then scroll down the page to the **Jira Software** section.
3. Under **Jira Software configuration**, select the **Relevant sprints** checkbox.
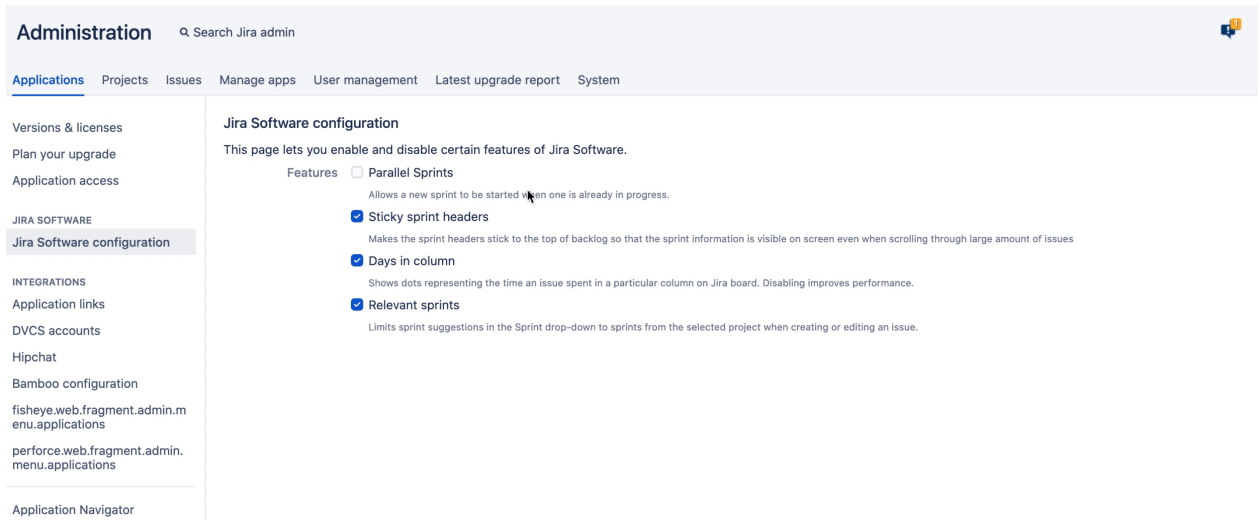
> ⓘ Users will still be able to assign sprints not related to any project without your clearing the checkbox by selecting the **Show all** checkbox in the Sprint drop-down.
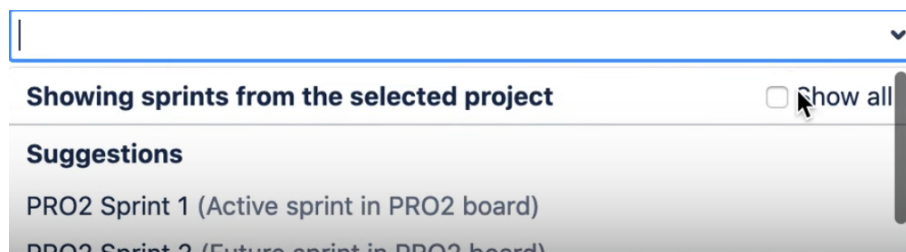
Releasing a version

Releasing a version is the culmination of your team's hard work. It may be the result of many iterations of work across a number of months. As the iteration manager, it will be your job to ensure that all the work is complete for the release, and to coordinate the activities needed to release the version.

As described in Configuring versions in a Scrum project and Configuring versions in a Kanban project, versions are handled differently in Scrum and Kanban projects. A version in a Scrum project is pre-planned and is released when the planned work is complete. In a Kanban project, a version can be released at any time — the version will contain all issues that are complete at that time.

The documentation in this section will help you release your version in Jira Software.

Search the topics in 'Releasing a version':

Overview

Check the release status of a version	Before you release a version, you need to be sure that everything is ready — issues are complete, code is committed, reviewed and merged, builds are passing, etc. In Jira Software, the Release Hub can provide you with the information you need for a release, in one place. Learn more: Checking the release status of a version
Deploy the release	Deploying your release requires actions to be taken on a number of different systems. The version in Jira Software needs to be released, build(s) need to be run to generate the artifact, the artifact needs to be deployed to the right environments, etc. Learn more: Deploying a release
Create the release documentation	Once you have deployed your release, you may want to create documentation to accompany it. If you have Jira Software connected to Confluence, you can generate a 'Change Log' report in Confluence for your stakeholders. Learn more: Jira Report Blueprint

Already released your version?

Next: Reporting

Project lifecycle
NEW PROJECT　　　BACKLOG　　　NEW VERSION　　　WORK　　　RELEASE　　　REPORTING
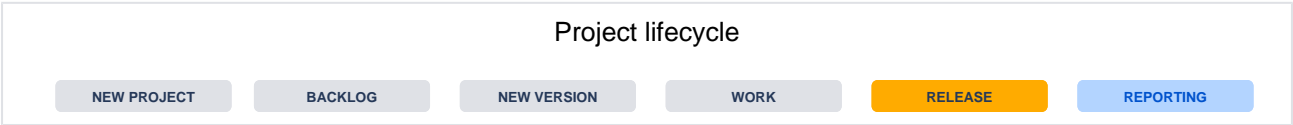
Checking the release status of a version

A version has many moving parts. As an iteration manager, you need to be sure that everything is ready before you can release — issues are complete, code is committed, reviewed and merged, builds are passing, etc. The Release Hub in Jira Software can help you stay on top of everything when it's time to release a version. Each issue in Jira Software also has a Development panel that helps you see the development status of an issue at a glance.

On this page:

- Using the Release Hub
- Checking the development status of an issue
- Next steps

Using the Release Hub

As described in Checking the progress of a version, the Release Hub is useful for seeing the status of a version at a point in time. It shows a summary and a breakdown of the issues in the version, grouped by status, so you can see if there's any remaining work for the release.

If you have Jira Software connected to your development tools, the 'Warnings' tab can also help you spot issues that could cause problems for the release. These include issues that should be in the release but haven't been merged (open pull requests), issues that haven't had code reviews, etc. Rather than spend your time hunting this information down, you can help your team sort these problems out as soon as they occur.

1. Navigate to the project that your version is in.
2. Click **Releases** > your desired version.
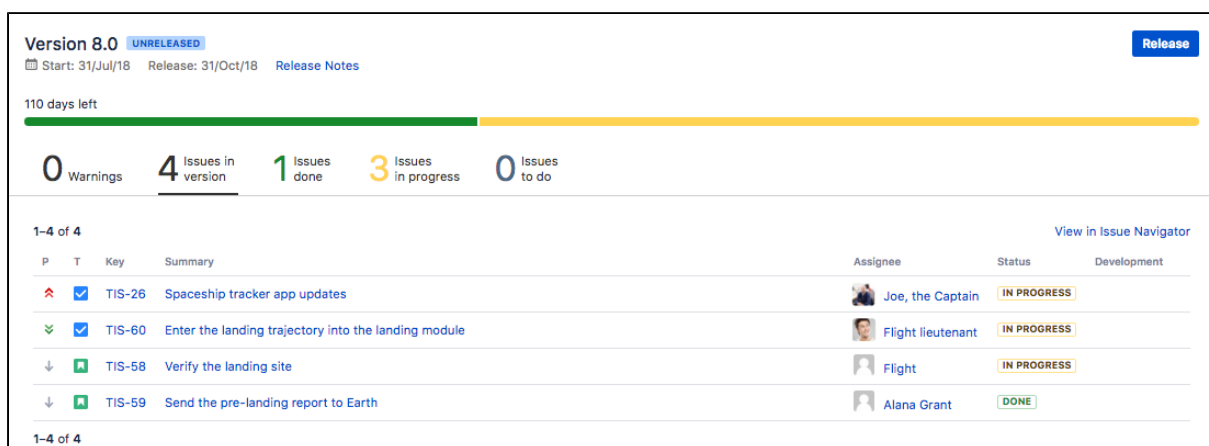
Checking the development status of an issue

If you need to deep dive into individual issues, the Development panel on each issue will show you its development status. This includes commits, builds, branches, pull requests, and more. For more information, see Viewing the development information for an issue.

Next steps

> **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Configuring development tools

Deploying a release

It's time to turn your team's hard work into a software release. By this stage, you should be confident that your version is ready to be released — issues are complete, code is checked in, reviewed and merged, builds are passing, etc.

To deploy a release, you would typically release the version in Jira Software, build the release, then deploy the release to the required environment.

Before you begin

- You must be a project administrator to do the Jira Software tasks on this page.

Release your version in Jira Software

Before you release your version, check the release status of your version.

Scrum

In a Scrum project, a version is created when you start work on the project. Issues are assigned to the version throughout development. Releasing the version is just a matter of marking the version as released.

1. Navigate to the project that your version is in.
2. Click **Releases** > your desired version.
3. On the version page, click the **Release** button.
4. The release dialog will be displayed.
5. Enter the details for the release, then click the **Release** button.

Note, if there are unresolved issues, you can choose to ignore these issues and proceed with the release, or move them to a later version.

Releasing a version on a Scrum board marks the version as released. It doesn't automatically modify the issues in the version in any way.

Kanban

In a Kanban project, you create the version at the time of the release. The version will contain all issues that are 'Done' (i.e. in the right-most column) at that time.

1. Navigate to the project that your version is in.
2. Click **Kanban board**.
3. Click **Release...**
4. The release dialog will be displayed.
5. Enter the details for the release, then click the **Release** button.

Releasing a version on a Kanban board does the following:

- For all issues that are 'Done', sets the Fix Version to the version that you just created.
- Marks the version as released.

As a result, the issues should disappear from the board, as the default Work Sub-Filter for a Kanban board only shows issues that have no Fix Version, or issues with a Fix Version that is unreleased. Note, if an issue has multiple Fix Versions, it will only disappear from the board once all versions are released.

Build the release

The next step is to build the release. A release is essentially a snapshot of the artifact(s) and related metadata that will be deployed. A release is created from a single build.

If you have connected Jira Software to Bamboo, you can automatically run a Bamboo build when you release a version. Otherwise, if you haven't connected Jira Software to Bamboo, or you are using a different build tool, you will need to run your builds independently.

Learn more: Running a Bamboo build when releasing a version

Deploy the release

Once you have built your artifacts, you need to deploy them to the required environment. You may do a few deployments to different environments (e.g. test, staging, etc) before releasing it to your customers.

If you are using Bamboo, deployment projects make it simple for you to deploy your artifacts to different environments. Otherwise, you will need to manually copy the files across environments.

Learn more: Deployment projects

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Configuring development tools

Running a Bamboo build when releasing a version

Releasing a new version of software usually involves a number of tasks, such as releasing the version in Jira Software, building and testing, merging code, creating tags, creating branches, labeling builds, etc. If you have connected Jira Software to Atlassian's Bamboo, you can trigger these tasks to run automatically at the release of a version in Jira Software.

When you release a Jira Software version, you will have the option of selecting a Bamboo plan and specifying which stages in the plan to run. Releasing the version will run the plan in Bamboo. If the Plan is successful, the version will be released on Jira Software. Otherwise, the version will not be released.

On this page:

- Before you begin
- Running a Bamboo build when releasing a version
- Next steps

Before you begin

- Your Jira administrator must have integrated Jira Software with Bamboo. For instructions, see Configuring development tools.
- Your Jira administrator must have installed the latest JIRA Bamboo app to use the release management feature. For instructions on how to install an app, see Managing apps.
- You must be a project administrator for the project that the version is in. See Permissions overview for more information.

Running a Bamboo build when releasing a version

1. Navigate to the project that your version is in.
2. Click **Releases**.
3. Click your desired version.
4. On the version page, click the **Release** button.
5. The release build dialog will be displayed.
6. Enter the build details for the release:

   No Build
   Choose this option if you *do not* want to run a Bamboo build, i.e. you only want to release the version in Jira Software .

   New Build
   Choose this option, if you want to run a Bamboo build that has not been started:
   * **Using Plan** — You can select any plan in the linked Bamboo instance that you have permission to view (unless your administrator has configured basic HTTP authentication). You'll also be able to choose a branch of your plan.
   * **Stages** — You can select the stages that you want to run for this release. Note, you cannot skip stages.
   * **Build Variables** — You can override any global variables or plan variables with your own parameters. See Running a plan build manually in Bamboo.

   Existing Build
   Choose this option, if you want to run a Bamboo build that is in progress and has been paused at an optional stage:
   * **Using Plan** — You can select any plan in the linked Bamboo instance that you have permission to view (unless your administrator has configured basic HTTP authentication). You'll also be able to choose a branch of your plan.
   * **Stages** — You can select the stages that you want to run for this release. Note, you cannot re-run stages that have been completed nor skip stages.
   * **Build Variables** — You can override any global variables or plan variables with your own parameters. See Running a plan build manually in Bamboo.
7. Click **Release**. The Bamboo build will run. If it is successful, the version will be released. If not, you can choose to run it again, or select a different plan.

Next steps

> (i) **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Configuring development tools

Working in an agile project

If you are a team member working with Jira Software, you are in the right place! These pages will be useful to you, if you are a developer, designer, person, technical writer, or anyone else that uses (but does not configure) *Jira Software*. The topics in this chapter will help you use Jira Software to manage your work.

In the wrong place? See the following chapters instead:

- *If you are responsible for running an agile project with Jira Software, see Leading an agile project.*
- *If you are an administrator installing or configuring the Jira Software server, see Administering Jira Software.*

Where do I start?

- If this is the first time you're using Jira Software, read our Getting started as a Jira Software user before you start reading other topics.
- If you're already familiar with Jira Software, use the search box below to find the topic you need.

Search the topics in 'Developing in an agile project':

Overview

The topics in this chapter cover the activities that you will need to do at any given time when you start using Jira Software.

> ⊘ You'll be working with boards in an agile project, and it may be worthy to know how to use the project sidebar in Jira Software. See Using the project sidebar for more information.

|	
Searching for issues	While you're doing development work in Jira Software, there will come a time when you'll have to search for issues. The issue you're currently working on may be related to an issue your team worked on before, and you want to find that issue. There are many ways for you to search for issues in Jira Software. Learn more Learn more...
Working with issues	There's so much that you can do in Jira Software when you start working on your issues. You can create issues, estimate how long it takes to complete issues, collaborate with your team while working on issues, reference issues to and from your source repository, view the development information of issues, etc. Learn more...
Configuring dashboards	You can add, configure, and customize dashboards in Jira Software. With gadgets, you can choose the information that appears on your dashboard or wallboard — if you're using dashboards for that purpose. You can use the gadgets that help keep you and your team at your most productive. Learn more...
Managing your user profile	You may want to personalize some details in your user profile as you go about your development work. You can edit your user details, change your avatar or your password, set your Jira homepage, manage your email notifications, and view and manage your OAuth tokens, etc. Learn more...

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. Any user 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 you how to run a search and use the search results. If you want more details on anything described on this page, see the related topics at the bottom of the page.

On this page:

- 1. Define your search criteria
- 2. Change your view of the search results
- 3. Working with the search results
- 4. Save your search
- Good to know
- Next steps

## 1. Define your search criteria

The first step in searching for issues is to define the criteria for your new search. You can define your search criteria in three different ways: using the quick search, using the basic search, or using the advanced search.

Quick search	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.
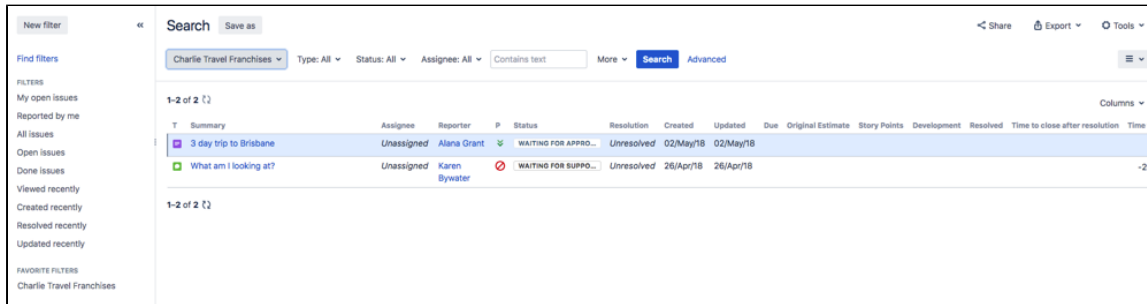
3. Working with the search results

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

Screenshot: Basic search
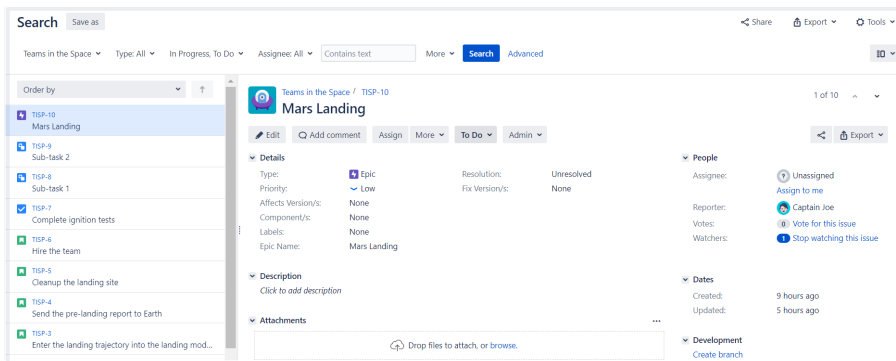
Basic searching

1. Choose **Issues** > **Search for issues**.
   - If there are existing search criteria, click the **New filter** button to reset the search criteria.
   - If the advanced search is shown instead of the basic search, click **Basic** (next to the **Search** button).

     In general, a query created using basic search will be able to be translated to advanced search, and back again. However, a query created using advanced search may not be able to be translated to basic search, particularly if:

     - the query contains an OR operator (note you can have an IN operator and it will be translated, e.g. `project in (A, B)`)
       - i.e. even though this query: `(project = JRA OR project = CONF)` is equivalent to this query: `(project in (JRA, CONF))`, only the second query will be translated.
     - the query contains a NOT operator
     - the query contains an EMPTY operator
     - the query contains any of the comparison operators: !=, IS, IS NOT, >, >=, <, <=
     - the query specifies a field and value that is related to a project (e.g. version, component, custom fields) and the project is not explicitly included in the query (e.g. `fixVersion = "4.0"`, without the `AND project=JRA`). This is especially tricky with custom fields since they can be configured on a Project/Issue Type basis. The general rule of thumb is that if the query cannot be created in the basic search form, then it will not be able to be translated from advanced search to basic search.
2. Enter the criteria for the search. You can search against specific fields and/or search for specific text.

   - If you are searching against a field and can't find the field you want, or the field is displaying greyed out text, see the Troubleshooting section below.
   - If you are searching for text, you can use special characters and modifiers in your search text, such as wildcards and logical operators. See Search syntax for text fields.
3. The search results will automatically update in the issue navigator, unless your administrator has disabled automatic updates of search results. If so, you will need to click the **Update** button on the field drop-down after every change.
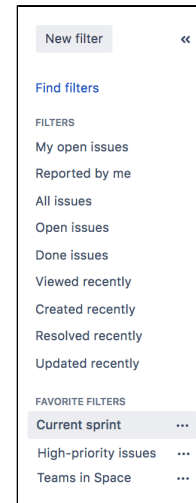
Running a saved search

Saved searches (also known as filters) are shown in the left panel, when using basic search. If the left panel is not showing, hover your mouse over the left side of the screen to display it.

To run a filter, e.g. **My Open Issues**, simply click it. The search criteria for the basic search will be set, and the search results will be displayed.

Note, clicking the **Recently Viewed** filter will switch you to the advanced search, as the basic search cannot represent the `ORDER BY` clause in this filter.

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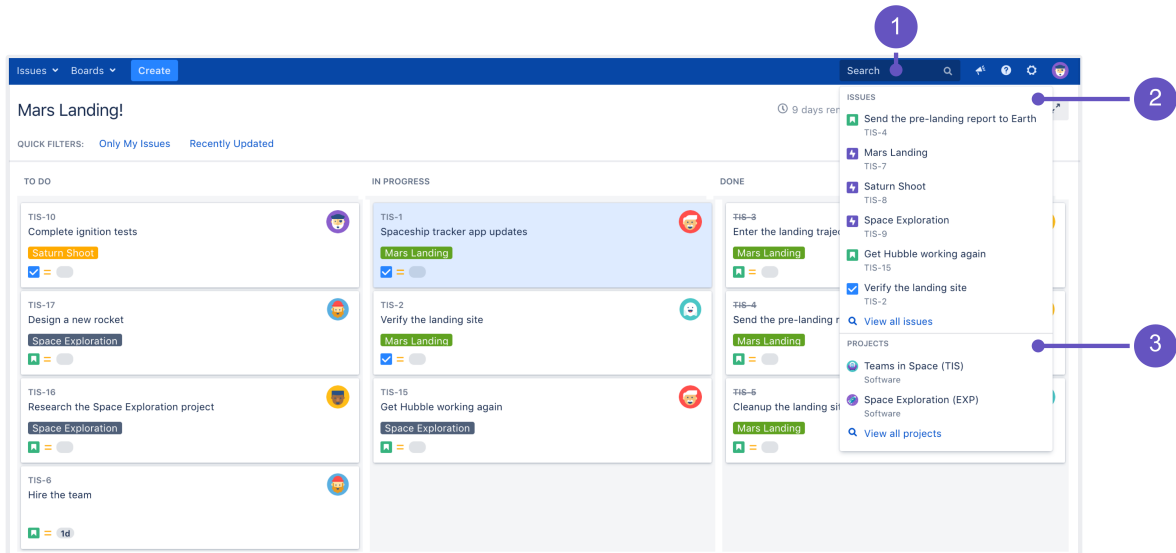
1. **Search:** Click anywhere in the box to display your recent work, or start typing to search through all your issues and projects.
2. **Issues:** Recent issues (before searching), or issues that match your search.
3. **Projects:** Recent projects (before searching), or projects that match your search.

> ✅ 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 you're looking for, the quick search will react instantly by showing and refreshing the list of most relevant results. To display these results, your search term is matched against the following fields:

- Summary (projects and issues)
- Description (issues)

Free-text searching

You can additionally search through comments or use extra operators for fuzzy or wildcard search. These results won't be displayed as 'instant results', but you can view them after pressing **Enter** in the search box.

You can combine free-text and keywords together, e.g. "`my closed test tasks`". You can also 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 the Issue Navigator, where you can view them in a variety of useful formats (Excel, XML, etc).

The search terms that quick search recognizes are:

Search Term	Description	Examples
`my`	Find issues assigned to me.	`my open bugs`
`r:`	Find issues reported by you, another user or with no reporter, using the prefix *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`

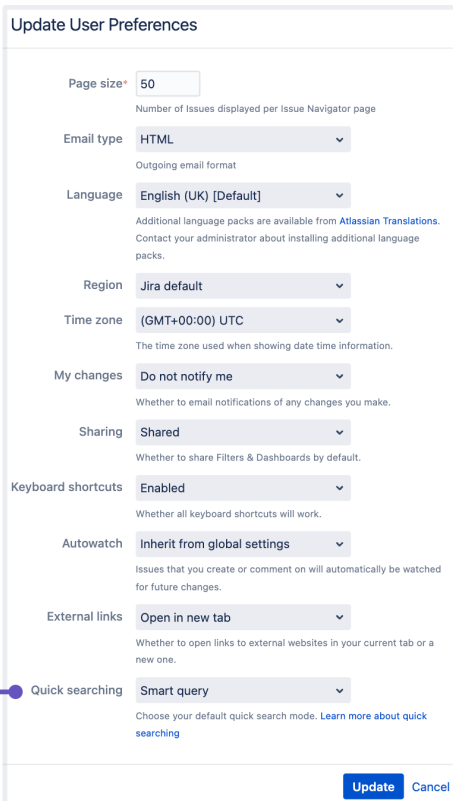| `<issue type>`	Find issues with a particular Issue Type. Note that you can also use plurals.	`bug`
		`task`
		`bugs`
		`tasks`

`<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

> (i) 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 don't 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 you're looking for.

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

Searching issues from your browser's search box

If you are using Firefox or Internet Explorer 8 (or later), you can add your Jira instance as a search engine /provider via the drop-down menu next to the browser's search box. Once you add your Jira instance as a search engine/provider in your browser, you can use it at any time to conduct a Quick Search for issues in that Jira instance.

> ⓘ **OpenSearch**
>
> Jira supports this browser search feature as part of the autodiscovery part of the OpenSearch standard, by supplying an OpenSearch description document. This is an XML file that describes the web interface provided by Jira's search function. Any client applications that support OpenSearch will be able to add Jira to their list of search engines.

Next steps

Read the following related topics:

- Searching for issues

Advanced searching

The advanced search allows you to build structured queries using the Jira Query Language (JQL) to search for issues. You can specify criteria that cannot be defined in the quick or basic searches. For example, you can use the ORDER BY clause to sort Jira issues either in descending or ascending order or narrow down your search results for the desired date range.

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

Before using the advanced search, consider the following:

- If you don't have complex search criteria, you may want to use quick search instead.
- If you are not comfortable with the Jira Query Language (JQL), you may want to use basic search instead.

> ℹ️ Note that JQL is not a database query language, even though it uses SQL-like syntax.

The following is the example of an advanced search query in Jira that returns all issues for the **Teams in the Space** project.

1. **JQL query** that refines the search results.
2. A list of Jira **issues** that match the search criteria.

Using advanced search

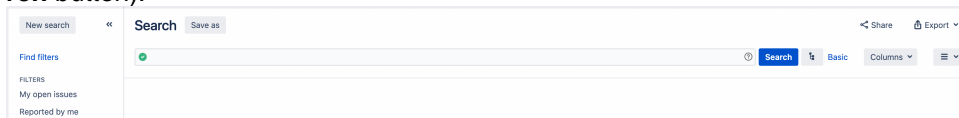1. Go to **Issues** (in the header) > **Search for issues**.

- If there are existing search criteria, select the **New search** button to reset the search criteria.

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

> ⓘ 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 don't see auto-complete suggestions, this might happen because of the following:

   - Your administrator may have disabled the "JQL Auto-complete" feature for your Jira instance.
   - Auto-complete suggestions are not offered for function parameters.
   - Auto-complete suggestions are not offered for all fields. Check the fields reference to see which fields support auto-complete.
3. Press Enter or select **Search** to run your query. Your search results will display in the issue navigator.

Switch between basic and advanced search

In general, a query created using basic search will be able to be translated to advanced search, and back again. However, sometimes a query created using an advanced search may not be able to be translated into a basic search. Expand the following section for details.

You might not switch between two searches if:

- the query contains an `OR` operator.

  > ⓘ 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 *captainjoe's* 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 wouldn't 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:

203

- `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 you're a Jira admin, note that this list is hard coded in the `JqlStringSupportImpl.java` file.

Performing text searches

You can use Lucene's text-searching features when performing searches on the following fields by using the `CONTAINS` operator.

- Summary.
- Description.
- Environment.
- Comments.
- Custom fields that use the "Free Text Searcher". These are custom fields of the following built-in custom field types: Free Text Field, Text Field, and Read-only Text Field.

When searching for text fields, you can also use single and multiple character wildcard searches. For more information, see Search syntax for text fields.

Differences between day and time search

A day (1d) and time (24h) values are differently calculated in a query and don't return the same results:

- If you specify "1d", the start of the day will start calculating at 00:00 of the server timezone unless you also add the exact time. "1d" will also include the current day if you execute the query now. It doesn't take into account the amount of time relative to the time you had executed the query (24 hours from the time you executed the JQL).
- If you use "24h", it will start calculating from the hour when you executed it (-24 hours from the time you run the JQL).

Example

Let's assume that you updated an issue's status to "Closed" yesterday at 3 PM. You run the following queries at 1 PM today:

- `status changed to "Closed" after -1d` won't return the closed issue. However, it'll return the result if you run `status changed to "Closed" after -2d`.
- `status changed to "Closed" after -24h` will return the closed issue.

Reference

Here you can find a brief overview of Jira fields, operators, keywords, and functions used to compose JQL queries. For detailed description   and examples of their usage for advance searching, check the links from the **Reference** column.

	Description	Reference

|		
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. • `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 a field on its left with one or more values (or functions) on its right, such that only true results are retrieved by the clause. Some operators may use the NOT keyword.	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.
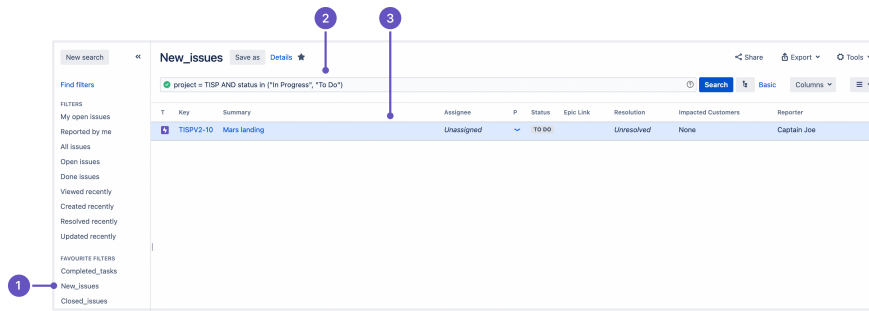
1. A **search saved as a filter**, which returns issues based on the criteria specified in a JQL query.
2. **JQL query** that specifies search criteria.
3. **Issues** that match the search criteria.

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 `/Installat ion-directory/atlassian-jira/WEB-INF/lib` and locate the Lucene jar files. The Lucene version number will be part of the filename.

209

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

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

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:

214

```
"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

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 field name or custom field ID (i.e. the number that Jira automatically allocates to an custom field).

Note, it is safer to search by custom field ID than by custom field name. It is possible for a custom field to have the same name as a built-in Jira system field; in which case, Jira will search for the system field (not your custom field). It is also possible for your Jira administrator to change the name of a custom field, which could break any saved filters that rely on that name. Custom field IDs, 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 syntax *can 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 different operators.*
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 different functions.*
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 syntax can be used.

Syntax	`environment`
Field Type	TEXT
Auto-complete	No
Supported operators	`~ , !~` `IS , IS NOT`

Unsupported operators	`= , != , > , >= , < , <=` `IN , NOT IN, WAS, WAS IN, WAS NOT, WAS NOT IN , CHANGED`
Supported functions	None
Examples	- Find issues where the environment contains text that matches "Third floor" (i. e. a "fuzzy" match): `environment ~  "Third floor"` - Find issues where the environment contains the exact phrase "Third floor": `environment ~  "\"Third floor\""`

^ top of page

Epic link

Search for issues that belong to a particular epic. The search is based on either the epic's name, issue key, or issue ID (i.e. the number that Jira automatically allocates to an issue).

Syntax	`"epic link"`
Field Type	Epic Link Relationship
Auto-complete	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`

^ top of page

Filter

You can use a saved filter to narrow your search. You can search by filter name or filter ID (i.e. the number that Jira automatically allocates to a saved filter).

Note:

- It is safer to search by filter ID than by filter name. It is possible for a filter name to be changed, which could break a saved filter that invokes another filter by name. Filter IDs, however, are unique and cannot be changed.
- An unnamed link statement in your typed query will override an ORDER BY statement in the saved filter.
- You cannot run or save a filter that would cause an infinite loop (i.e. you cannot reference a saved filter if it eventually references your current filter).

Syntax	`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 =  12000 and 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 the version order that has been set up by your project administrator, not a numeric or alphabetic order. *For this field, the contain operators (~ and !~) find exact matches, and can be used to search through versions with a wildcard.*

Unsupported operators	
Supported functions	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 issue ID (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 team's 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`
	ⓘ 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 security name or issue level security ID (i.e. the number that Jira automatically allocates to an issue level security).

Note, it is safer to search by security level ID 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 priority name or priority ID (i.e. the number that Jira automatically allocates to a priority).

Note, it is safer to search by priority ID than by priority name. It is possible for your Jira administrator to change the name of a priority, which could break any saved filter that rely on that name. Priority IDs, however, are unique and cannot be changed.

Syntax	`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 project name, by project key or by project ID (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`

^ top of page

Remaining estimate

Only available if time-tracking has been enabled by your Jira administrator.

Search for issues where the remaining estimate is set to a particular value (i.e. a number, not a date or date range). Use "w", "d", "h" and "m" to specify weeks, days, hours, or minutes.

Syntax	`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

^ top of page

Reporter

Search for issues that were reported by a particular user. This may be the same as the creator, but can be distinct. You can search by the user's full name, ID, or email address.

Syntax	`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

229

- 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.)

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`

Request last activity time

Only applicable if Jira Service Management is installed and licensed.

Search for issues that were last acted on or created:

- on a particular date.
- before/after a particular date (or date range).

Note that if a time-component is not specified, midnight will be assumed. Please note that the search results will be relative to your configured time zone (which is by default the Jira server's time zone).

Use one of the following formats:

230

```
"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"`

^ top of page

Resolution

Search for issues that have a particular resolution. You can search by resolution name or resolution ID (i.e. the number that Jira automatically allocates to a resolution).

Note, it is safer to search by resolution ID than by resolution name. 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

^ top of page

Resolved

Search for issues that were resolved on, before, or after a particular date (or date range). Note that if a time-component is not specified, midnight will be assumed. Please note that the search results will be relative to your configured time zone (which is by default the Jira server's time zone).

Use one of the following formats:

`"yyyy/MM/dd HH:mm"`
`"yyyy-MM-dd HH:mm"`
`"yyyy/MM/dd"`
`"yyyy-MM-dd"`

Or use `"w"` (weeks), `"d"` (days), `"h"` (hours) or `"m"` (minutes) to specify a date relative to the current time. The default is `"m"` (minutes). Be sure to use quote-marks (`"`); if you omit the quote-marks, the number you supply will be interpreted as milliseconds after epoch (1970-1-1).

Syntax	`resolved`
Alias	`resolutionDate`
Field Type	DATE
Auto-complete	No
Supported operators	`= , != , > , >= , < , <=` `IS , IS NOT, IN , NOT IN`

Unsupported operators	`~ , !~` `WAS, WAS IN, WAS NOT, WAS NOT IN , CHANGED`
Supported functions	When used with the EQUALS, NOT EQUALS, GREATER THAN, GREATER THAN EQUALS, LESS THAN or LESS THAN EQUALS operators, this field supports: • 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()

233

- 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 status name or status ID (i.e. the number that Jira automatically allocates to a status).

Note:

- It is safer to search by status ID than status name. 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")`

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)
  - Text field (< 255 characters)
  - Read-only text field

Notes:

- The `text` master-field can only be used with the CONTAINS operator ("~").
- Jira text-search syntax can be used with these fields.

Syntax	`text`
Field Type	TEXT
Auto-complete	No
Supported operators	**~**
Unsupported operators	**= , != , !~ , > , >= , < , <=** **IS , IS NOT , IN , NOT IN , WAS, WAS IN, WAS NOT, WAS NOT IN , CHANGED**
Supported functions	None

Examples	- 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\""`

^ top of page

Time spent

Only available if time-tracking has been enabled by your Jira administrator.

Search for issues where the time spent is set to a particular value (i.e. a number, not a date or date range). Use "w", "d", "h" and "m" to specify weeks, days, hours, or minutes.

Syntax	`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	

237

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

Type

Search for issues that have a particular issue type. You can search by issue type name or issue type ID (i.e. the number that Jira automatically allocates to an issue type).

Note, it is safer to search by type ID than type name. 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`

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")`

^ top of page

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`

241

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")`

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 created on or before 12th December 2010 (but not 13th December 2010):
`worklogDate <= "2010/12/13"`
Find issues that have comments in work log entries created on 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 created less 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. Work ratio is 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.

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
```

> ⓘ 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).

247

Advanced searching - development fields reference

If you've linked your development tools to Jira, you can use even more parameters when performing a search. A
dding development information to your JQL queries gives you more visibility of your development tools in Jira.
So, you'll make better-informed decisions on your development and work, build a better, more reliable release
pipeline, and deliver higher quality releases to your happy customers.

JQL is made up of clauses. In a clause, a field is followed by an operator. The operator is followed by one or
more values or functions. The operator compares the value of the field on the left side with one or more values
or functions on the right side. So, only valid results are retrieved by the clause. But you can't compare two fields
in JQL.

To use development fields in your JQL queries, you can enter something like this when doing an advanced
search:

```
fixVersion = 7.8.0 AND development[pullrequests].open > 0
```

This query will return all issues with the fix version 7.8.0 and open pull requests linked to them.

And what if you want to know how many builds are failing in your project? Try this query:

```
project = ABC AND development[builds].failing > 0
```

> ⓘ When using development fields for advanced search, you may see an error stating that the field doesn't
> exist or you don't have permission to view it. Learn more about this issue and how to resolve it in our kn
> owledge base article.

Now, check the following development fields that can be used in JQL queries.

development[branches].all

Search for issues that have branches on Bitbucket instances.

Syntax	`development[branches].all`
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 all issues with more than two branches: `development[branches].all > 2` - Find all issues with at least one branch: `development[branches].all > 0`

^ top of page

development[builds].all

Search for issues that have builds on linked Bamboo instances.

Syntax	`development[builds].all`
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 all issues with more than four builds: `development[builds].all > 4` - Find all issues with at least one build: `development[builds].all > 0`

^ top of page

development[builds].failing

Search for issues that have failed builds on linked Bamboo instances. Note that these issues may also have passed builds associated with them, so create your JQL query to exclude passed builds if that's what you require.

Syntax	`development[builds].failing`
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 all issues with more than three failing builds: `development[builds].failing > 3`

^ top of page

development[builds].passing

Search for issues that have passed builds on linked Bamboo instances. Note that these issues may also have failed builds associated with them, so create your JQL query to exclude failed builds if that's what you require.

Syntax	`development[builds].passing`
Field Type	TEXT
Auto-complete	No
Supported operators	`= , != , > , >= , < , <=`

249

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

Unsupported operators	`~ , !~, IS , IS NOT , IN , NOT IN, WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED`
Supported functions	None
Examples	Find all issues with more than one passed build: `development[builds].passing > 1`

development[builds].status

Search for issues that have builds that have passed or failed on linked Bamboo instances.

Syntax	`development[builds].failing`
Field Type	TEXT
Auto-complete	No
Supported operators	`= , != , IN , NOT IN`
Unsupported operators	`> , >= , < , <=, ~ , !~, IS , IS NOT , WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED`
Supported functions	passed, failed

Examples	• Find all issues with only passed builds:
	`development[builds].status in (passed)`
	• Find all issues with only failed builds:
	`development[builds].status in (failed)`
	• Find all issues with any information on builds:
	`development[builds].status in (failed, passed)`

development[commits].all

Search for issues that have code commits on linked Bitbucket instances.

Syntax	`development[commits].all`
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 all issues with more than 15 commits: `development[commits].all > 15` - Find all issues with at least one commit: `development[commits].all > 0`

development[deployments].all

Search for issues that have successful and unsuccessful deployments on Bamboo instances.

Syntax	`development[deployments].all`
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 all issues with less than ten deployments: `development[deployments].all < 10` - Find all issues with at least one deployment: `development[deployments].all > 0`

development[deployments].deployed

Search for issues that have successful deployments on linked Bamboo instances.

Syntax	`development[deployments].deployed`
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 all issues with more than two successful deployments: `development[deployments].deployed > 2`

development[deployments].environment

Search for issues that have deployments to a specific environment on linked Bamboo instances. The supported environment functions are set in the Bamboo instances by the instance admin and can vary.

Syntax	`development[deployments].environment`
Field Type	TEXT
Auto-complete	No
Supported operators	`= , != , IN , NOT IN`
Unsupported operators	`> , >= , < , <=, ~ , !~, IS , IS NOT ,` `WAS, WAS IN, WAS NOT, WAS NOT IN, CHANGED`
Supported functions	The supported environment functions are set in the Bamboo instances by the instance admin and can vary.

Examples	Find all issues with deployments to a production environment:
	`development[deployments].environment in (production)`
	`development[deployments].environment = production`

^ top of page

development[pullrequests].all

Search for issues that have open or merged pull requests on linked Bitbucket instances.

Syntax	`development[pullrequests].all`
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 all issues with more than five open or merged pull requests: `development[pullrequests].all > 5` - Find all issues with at least one pull request: `development[pullrequests].all > 0`

^ top of page

development[pullrequests].declined

Search for issues that have declined pull requests on linked Bitbucket instances. Note that this query will also return issues with merged pull requests.

Syntax	`development[pullrequests].declined`
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 all issues with more than three declined pull requests: `development[pullrequests].declined > 3`

^ top of page

development[pullrequests].merged

Search for issues that have merged pull requests on linked Bitbucket instances. Note that this query will also return issues with declined requests.

Syntax	`development[pullrequests].merged`
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 all issues with more than five merged pull requests: `development[pullrequests].merged > 5`

^ top of page

development[pullrequests].open

Search for issues that have open pull requests on linked Bitbucket instances.

Syntax	`development[pullrequests].open`
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 all issues with more than two open pull requests: `development[pullrequests].open > 2`

^ top of page

development[pullrequests].status

Search for issues that have open or merged pull requests on linked Bitbucket instances.

Syntax	`development[pullrequests].status`
Field Type	TEXT
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 with open pull requests:
	`development[pullrequests].status = open`
	`development[pullrequests].status in (open)` - Find all issues with open and merged pull requests:
	`development[pullrequests].status in (open,merged)`

^ top of page

development[reviews].all

Search for issues that have open or closed reviews on linked Crucible instances.

Syntax	`development[reviews].all`
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 all issues with more than two open or closed reviews: `development[reviews].all > 2` - Find all issues with at least one review: `development[reviews].all > 0`

^ top of page

development[reviews].open

Search for issues that have open reviews on linked Crucible instances.

Syntax	`development[reviews].open`

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 all issues with open reviews: `development[reviews].open > 0`

^ top of page

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
```

256

- Find all issues that are not assigned to `jsmith`:

```
assignee != jsmith or assignee is empty
```

- Find all issues that were reported by you but aren't assigned to you:

```
reporter = currentUser() and assignee != currentUser()
```

- Find all issues where the Reporter or Assignee is anyone except John Smith:

```
assignee != "John Smith" or reporter != "John Smith"
```

- Find all issues that aren't unassigned:

```
assignee is not empty
```

or

```
assignee != null
```

^top of page

GREATER THAN: >

The `>` operator is used to search for issues where the value of a specified field is greater than a specified value.

> ⓘ The operator can only be used with fields that support ordering and can't be used with text fields. For example, date fields and version fields.
>
> To see a field's supported operators, check the individual field reference.

Examples

- Find all issues with more than four votes:

```
votes > 4
```

- Find all overdue issues:

```
duedate < now() and resolution is empty
```

- Find all issues where priority is higher than "Normal":

```
priority > normal
```

^top of page

GREATER THAN EQUALS: >=

The `>=` operator is used to search for issues where the value of a specified field is greater than or equal to a specified value.

> (i) The operator can only be used with fields that support ordering and can't be used with text fields. For example, date fields and version fields.
>
> To see a field's supported operators, check the individual field reference.

Examples

- Find all issues with four or more votes:

```
votes >= 4
```

- Find all issues due on or after 31/12/2008:

```
duedate >= "2008/12/31"
```

- Find all issues created in the last five days:

```
created >= "-5d"
```

^top of page

LESS THAN: <

The < operator is used to search for issues where the value of a specified field is less than a specified value.

> (i) The operator can only be used with fields that support ordering and can't be used with text fields. For example, date fields and version fields.
>
> To see a field's supported operators, check the individual field reference.

Examples

Find all issues with less than votes:

```
votes < 4
```

^top of page

LESS THAN EQUALS: <=

The <= operator is used to search for issues where the value of a specified field is less than or equal to a specified value.

> (i) The operator can only be used with fields that support ordering and can't be used with text fields. For example, date fields and version fields.
>
> To see a field's supported operators, check the individual field reference.

Examples

- Find all issues with four or fewer votes:

```
votes <= 4
```

- Find all issues that haven't been updated in the last month (30 days):

```
updated <= "-4w 2d"
```

IN

The `IN` operator is used to search for issues where the value of a specified field is one of multiple specified values. The values are specified as a comma-separated list, surrounded by parentheses.

> (i) Using `IN` is equivalent to using multiple `EQUALS`(`=`) statements, but is shorter and more convenient. That is, `reporter IN (tom, jane, harry)` is the same as `reporter = "tom" OR reporter = "jane" OR reporter = "harry"`.

Examples

- Find all issues that were created by either `jsmith`, `jbrown`, or `jjones`:

```
reporter in (jsmith,jbrown,jjones)
```

- Find all issues where the Reporter or Assignee is either Jack or Jill:

```
reporter in (Jack,Jill) or assignee in (Jack,Jill)
```

- Find all issues in version 3.14 or version 4.2:

```
affectedVersion in ("3.14", "4.2")
```

NOT IN

The `NOT IN` operator is used to search for issues where the value of a specified field isn't one of multiple specified values.

> (i) Using `NOT IN` is equivalent to using multiple `NOT_EQUALS`(`!=`) statements, but is shorter and more convenient. That is, `reporter NOT IN (tom, jane, harry)` is the same as `reporter != "tom" AND reporter != "jane" AND reporter != "harry"`.
>
> Also, the `NOT IN` operator won't match a field that has no value (a field is empty). For example, `assignee not in (jack,jill)` will only match issues that have an assignee and this assignee isn't "jack" or "jill".
>
> To find issues that are assigned to someone other than "jack" or "jill" or are unassigned, you should type: `assignee not in (jack,jill) or assignee is empty`.

Examples

- Find all issues where the Assignee is someone other than Jack, Jill, or John:

```
assignee not in (Jack,Jill,John)
```

- Find all issues where the Assignee isn't Jack, Jill, or John:

```
assignee not in (Jack,Jill,John) or assignee is empty
```

- Find all issues where the fix version isn't A, B, C, or D:

```
FixVersion not in (A, B, C, D)
```

- Find all issues where the fix version isn't A, B, C, or D, or has not been specified:

```
FixVersion not in (A, B, C, D) or FixVersion is empty
```

^top of page

> (i) 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"`.

> (i) 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:

- 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
```

262

IS NOT

The `IS NOT` operator can only be used with the EMPTY or NULL keywords. That is, it's used to search for issues where a specified field has a value.

> ⓘ Not all fields are compatible with this operator. For more details, see the individual field reference.

Examples

- Find all issues that have one or more votes:

```
votes is not empty
```

or

```
votes is not null
```

WAS

The `WAS` operator is used to find issues that currently have or previously had a specified value for a specified field.

In a search query, with this operator, you can use the following:

- `AFTER "date"`
- `BEFORE "date"`
- `BY "username"`
- `DURING ("date1","date2")`
- `ON "date"`

The `WAS` operator will match the value name (for example, "Resolved") that was configured in your system at the time when the field was changed.

The operator will also match the value ID associated with the value name. For example, it will match "4" as well as "Resolved".

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

Examples

- Find issues that currently have or previously had the status "In Progress":

```
status WAS "In Progress"
```

- Find issues that were resolved by Joe Smith before February 20:

```
status WAS "Resolved" BY jsmith BEFORE "2011/02/20"
```

- Find issues that were resolved by Joe Smith during 2010:

```
status WAS "Resolved" BY jsmith DURING ("2010/01/01","2011/01/01")
```

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, and Status.

Examples

- Find all issues that currently have or previously had the status "Resolved" or "In Progress":

  ```
  status WAS IN ("Resolved","In Progress")
  ```

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, and Status.

Examples

- Find issues that have never had the status "Resolved" or "In Progress":

```
status WAS NOT IN ("Resolved","In Progress")
```

- Find issues that didn't have the status "Resolved" or "In Progress" before February 20:

```
status WAS NOT IN ("Resolved","In Progress") BEFORE "2011/02/20"
```

> ⓘ 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 system at the time when the field was changed.

The operator will also match the value ID associated with the value name. For example, it will match "4" as well as "Resolved".

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

Examples

- Find issues that don't have and have never had the status "In Progress":

```
status WAS NOT "In Progress"
```

- Find issues that didn't have the status "In Progress" before February 20:

```
status WAS NOT "In Progress" BEFORE "2011/02/20"
```

CHANGED

The `CHANGED` operator is used to find issues where the value of a specified field was changed.

In a search query, with this operator, you can use the following:

- `AFTER "date"`
- `BEFORE "date"`

- `BY "username"`

- `DURING ("date1","date2")`

- `ON "date"`

- `FROM "oldvalue"`

- `TO "newvalue"`

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

Examples

- Find issues where the Assignee was changed:

  ```
  assignee CHANGED
  ```

- Find issues where the status was changed from "In Progress" back to "Open":

  ```
  status CHANGED FROM "In Progress" TO "Open"
  ```

- Find issues where the priority was changed by the user `freddo` after the start and before the end of the current week:

  ```
  priority CHANGED BY freddo BEFORE endOfWeek() AFTER startOfWeek()
  ```

^top of page

Advanced searching - functions reference

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

A function in JQL appears as a word followed by parentheses, which may contain one or more explicit values or Jira system fields. In a clause, a function is preceded by an operator, which in turn is preceded by a field. A function performs a calculation on either specific Jira data or the function's content in parentheses, such that only true results are retrieved by the function, and then again by the clause in which the function is used.

Some Jira apps can add additional functions to the advanced issue search. For example, the ScriptRunner for JIRA app 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

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 completed Sprint.

> ⓘ 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 completed sprint: `sprint in closedSprints()`

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)`

^ top of page

elapsed()

Only applicable if Jira Service Management is installed and licensed.

Returns issues whose clock is at a certain point relative to a cycle's start event.

Syntax	`elapsed()`
Supported fields	
Supported operators	`= , != , > , >= , < , <=`
Unsupported operators	`~ , IS , IS NOT , WAS , WAS IN , WAS NOT , WAS NOT IN , CHANGED`
Examples	Find issues that have been waiting for a first response for more than 1 hour: `"Time to First Response" > elapsed("1h")`

^ top of page

endOfDay()

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

Syntax	`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")`

^ top of page

endOfMonth()

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

Syntax	`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")`

^ top of page

endOfWeek()

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

For the `endOfWeek()` function, the result depends upon your locale. For example, in Europe, the first day of the week is generally considered to be Monday, while in the USA, it is considered to be Sunday.

274

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 a sprint 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()`

latestReleasedVersion()

Perform searches based on the latest released version (i.e. the most recent version that has been released) of a specified project. See also releasedVersions().

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

Syntax	`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)`

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")`

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()`

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.

> ⓘ 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 a sprint 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, use withincalendarhours().

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()`

^ top of page

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()`

^ top of page

pendingBy()

Only applicable if Jira Service Management is installed and licensed.

Search for issues that require approval by one or more of the listed users. This uses an `OR` operator, and you must specify the usernames.

Syntax	`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.

ⓘ 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 are not the reporter, and therefore don't have the permission specified. Also note, this function is only available if you are logged in to Jira.

Syntax	`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* para meter. See also latestReleasedVersion().

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 also startOfWeek(), startOfMonth(), startOfYear(), endOfDay(), endOfWeek(), endOfMonth(), and endOfYear().

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()

286

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 the Watcher 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()`

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()`

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 you're using advanced search, you need to additionally escape each of the quote-marks with a backslash (\). For details, see the examples below or find your field in Advanced search - field reference.

Examples

- **Basic search:** Find all issues that contain the phrase *Jira Software*:

  ```
  Just type "Jira Software" into the search field.
  ```

- **Advanced search:** Find all issues that contain the words *Jira* and *Software*, in no particular order.

  ```
  text ~ "Jira Software"
  ```

- **Advanced search:** Find all issues that contain the phrase *Jira Software*.

292

```
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 aren't related to your search query. The same pair of quote-marks would be automatically added by Jira in the basic search after running your search.

Using special characters to create phrases

In previous versions of Jira, you could use some special characters to combine **terms** into **phrases**, for example *Jira+Software* or *Jira/Software*. This is no longer the case, as the mechanism used for searching has changed and the special characters surrounding **terms** are ignored.

Wildcard searches: ? and *

Jira supports single and multiple character wildcard searches.

To perform a single character wildcard search, use the "`?`" symbol.

To perform a multiple character wildcard search, use the "`*`" symbol.

> ⓘ Wildcard characters need to be enclosed in quote-marks, as they are reserved characters in advanced search. Use quotations, e.g. `summary ~ "cha?k and che*"`

The single character wildcard search looks for terms that match that with the single character replaced. For example, to search for "`text`" or "`test`", you can use the search:

```
te?t
```

Multiple character wildcard searches 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` m ay exist.

> (i) Do not use the grouping character '(' at the start of a search query, as this will result in an error. For example, "`(atlassian OR Jira) AND bugs`" will not work.

Special characters

```
+ - & | ! ( ) { } [ ] ^ ~ * ? \ :
```

Special characters aren't stored in the index, which means you can't search for them. The index only keeps text and numbers, so searching for "`\\[Jira Software\\]`" and "`Jira Software`" will have the same effect — escaped special characters (`[ ]`) won't be included in the search.

In previous Jira versions, you could use special characters to combine two separate terms into a phrase, for example "`Jira+Software`" or "`Jira/Software`". This doesn't apply to Jira 8.x. If you'd like to search for phrases, see Exact searches (phrases).

Reserved words

To keep the search index size and search performance optimal in Jira, the following English *reserved words* (also known as '*stop words*') are ignored from the search index and hence, Jira's text search features:

```
"a", "an", "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.

ℹ 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
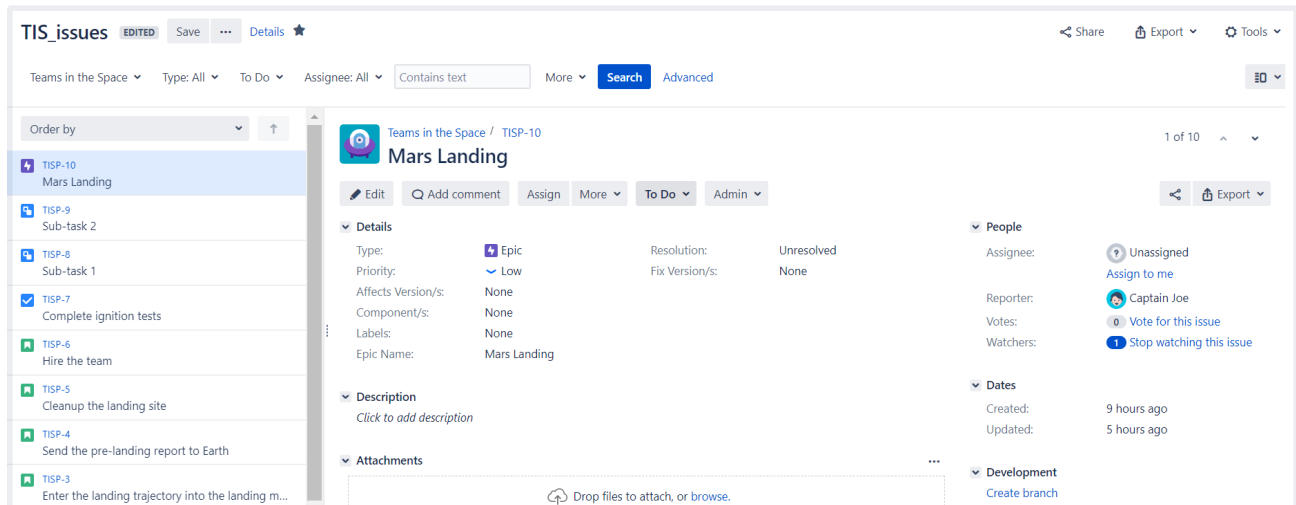
Saving a search as a filter

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, either clone (aka copy) the shared filter, or ask your Jira administrator to change the filter's ownership.

Update the filter's details:

1. Click the **My** tab on the 'Manage Filters' page.
2. Locate the filter you wish to update, click the **cog icon** > **Edit**.
3. The **Edit Current Filter** page displays, where you can update the filter details as required.
4. Click **Save** to save your changes.

> ⓘ 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)
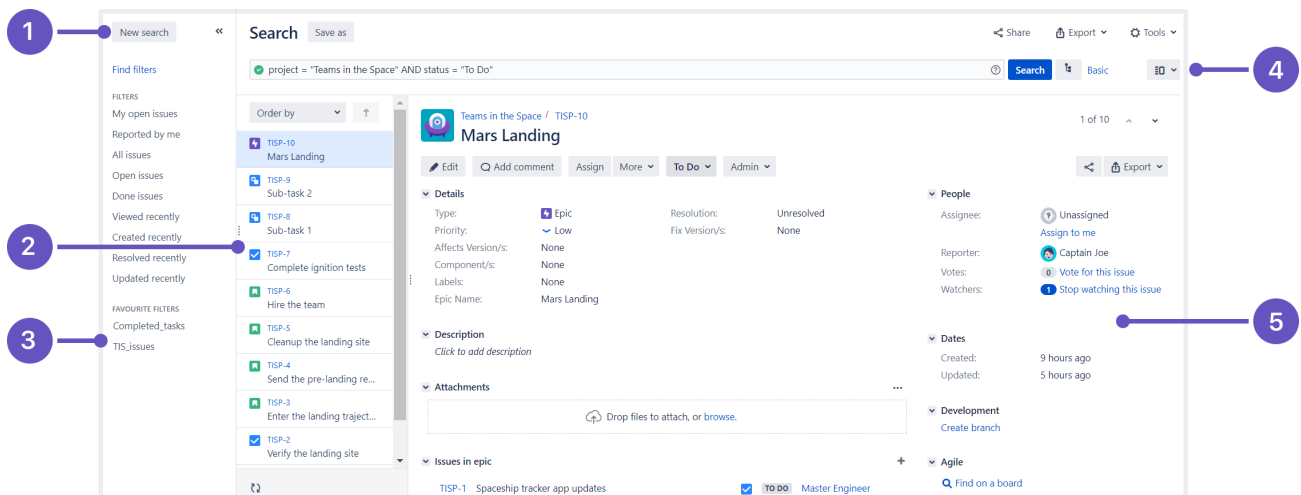
1. **Filter panel:** Click << to collapse the filter panel so you can have more space in the detail view.
2. **Issue:** Select an issue from this panel to see the details in the detail view window.
3. **Filters:** Select a filter to see all the matching issues in the panel to the immediate right.
4. **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 the sub-task aggregate columns (i.e. all columns beginning with ''). • If you sort the search results for an advanced search, an 'ORDER BY' clause will be added/updated for your JQL query to reflect the order of issues in your search results.
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 been hidden and that you have permissions to see. • It is possible to add any of the existing custom fields to the column list, as long as the fields are visible, and you have the right permissions. • Some custom fields, even if selected, do not appear in the Issue Navigator for all issues. For example, project-specific custom fields will be shown only if the filter has been restricted to that project only. Issue type custom fields will only appear if the filter has been restricted to that issue type.

Working with individual issues

You can action individual issues in your search results, directly from the issue navigator. Note that the list of issues will remain constant even if you change an issue, so that it doesn't meet the original search criteria. The advantage of this is that you have a constant set of search results that you can work from when triaging issues.

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 to email a link to a search result or shared filter.

- Recipients will receive an email with a link to the search result and the content of the **Note** field (if specified). The subject of the email will state that you (using your username) shared the issue.
- If you share the results of a filter, rather than an ad-hoc search, recipients will receive a link to the filter. Note, if the recipient does not have permission to view the filter, they will receive a link to the search results instead.

Displaying your search results in Confluence

If your Jira applications are connected to Confluence, you can display your search results on a Confluence page using the Jira issues macro. For instructions, see Jira issues macro.

Displaying your search results as a chart

Click **Export** > **Dashboard charts**. Choose the desired chart from the dialog that is displayed, then click **Save to Dashboard**.

The chart will be added to your dashboard. For more information on what each chart shows, see Reporting.

Exporting your search results

CSV	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 **resolutionda te**)	`<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`

308

| 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, see Editing multiple issues at the same time.

Next steps

Read the following related topics:

- Searching for issues
- Constructing cron expressions for a filter subscription

Constructing cron expressions for a filter subscription

This page describes how to construct a cron expression. Cron expressions can be used when creating a subscription to a filter, as described in Working with search results.

A cron expression gives you more control over the frequency, compared to the default schedules. For example, you could define a cron expression to notify you at 8:15 am on the second Friday of every month.

Constructing a cron expression

A cron expression is a string of fields separated by spaces. The following table displays the fields of a cron expression, *in the order that they must be specified (from left to right)*:

	Second	Minute	Hour	Day-of-month	Month	Day-of-week	Year (optional)
Allowed values	0-59	0-59	0-23	1-31	1-12 or JAN-DEC	1-7 or SUN-SAT	1970-2099
Allowed special characters	, - * /	, - * /	, - * /	, - * / ? L W C	, - * /	, - * / ? L C #	, - * /

Note, cron expressions are not case-sensitive.

Here is an example:

```
0 15 8 ? JAN MON 2014
```

This literally translates to 0 second, 15 minute, 8 hour, any day of the month, January, 2014.

In plain English, this represents 8:15am on every Monday during January of 2014. Note, the ? character means "no particular value". In this example, we've set the Day-of-month to no particular value. We don't need to specify it, as we've specified a Day-of-week value. Read more about special characters in the next section.

More examples of cron expressions are explained in the Examples section at the bottom of this page.

Special characters

Special character	Usage
,	Specifies a list of values. For example, in the **Day-of-week** field, 'MON,WED,FRI' means 'every Monday, Wednesday, and Friday'.
-	Specifies a range of values. For example, in the **Day-of-week** field, 'MON-FRI' means 'every Monday, Tuesday, Wednesday, Thursday and Friday'.
*	Specifies all possible values. For example, in the **Hour** field, '*' means 'every hour of the day'.
/	Specifies increments to the given value. For example, in the **Minute** field, '0/15' means 'every 15 minutes during the hour, starting at minute zero'.
?	Specifies no particular value. This is useful when you need to specify a value for one of the two fields **Day-of-month** or **Day-of-week,** but not the other.

L	Specifies the last possible value; this has different meanings depending on context. In the **Day-of-week** field, 'L' on its own means 'the last day of every week' (i.e. 'every Saturday'), or if used after another value, means 'the last xxx day of the month' (e.g. 'SATL' and '7L' both mean 'the last Saturday of the month'). In the **Day-of-month** field, 'L' on its own means 'the last day of the month', or 'LW' means 'the last weekday of the month'.
W	Specifies the weekday (Monday-Friday) nearest the given day of the month. For example, '1W' means 'the nearest weekday to the 1st of the month' (note that if the 1st is a Saturday, the email will be sent on the nearest weekday *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.

Working with issues

Need help working with issues? In Jira Software, you can create issues, estimate issues, even manage your code via issues. On this page, you'll find a quick overview for everything that you can do with an issue, such as creating issues, updating issues with estimates, creating branches in your source repository that reference the relevant issues in Jira Software, viewing development work for issues, resolving issues, 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.

Issues are the building blocks of any Jira project. An issue could represent a story, a bug, a task, etc in your project. This is what an issue looks like in Jira Software:

On a board (Scrum project — backlog)

1. Selected issue.
2. Issue actions.
3. Issue detail view.

Full view (e.g. via search results)

1. Project, key, and summary.
2. Issue actions, including workflow transitions menu.
3. Activity tabs that feature all changes related to the issue.

4. Share and export issue.
5. Development panel.
6. Agile panel.

> (i) The **issue detail view** may show different fields to the full view if it has been customized by your administrator. To see the full view, open the issue in a new tab/window (e.g. click the issue key with your middle mouse button).

Most information on an issue is self-explanatory. Here are a few things to be aware of though:

This panel only shows if Jira is connected to your development tools. You will be able to see related commits, builds, etc, to help you evaluate the development status of your issue. See Viewing the development information for an issue.
This panel will be shown on the issue, if your administrator has set Time Tracking to 'Remaining Estimate and Time Spent'. If an issue has sub-tasks, the Remaining estimate for the issue plus sub-tasks is rolled up into the parent issue.
The **History** tab of an issue records the following information: creator of the issue (this may be the same as the reporter, but can be distinct), changes to an issue field, attachment of a file, deletion of a comment, deletion of a worklog, creation or deletion of an issue link.

The **Activity** tab has the same information, plus additional information, such as comments. However, this may load more slowly, especially if there has been a lot of activity on the issue.

Next steps

Check out the following pages to reach issue ninja status:

- Creating issues and sub-tasks
- Attaching files and screenshots to issues
- Editing and collaborating on issues
- Logging work on issues

Attaching files and screenshots to issues

To share information with your 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.

On this page:

- Before you begin
- Add attachments
- Sort and manage attachments
- Access ZIP file contents
- Capture and attach screenshots

Before you begin

A Jira admin must enable specific user permissions so that you can add attachments and screenshots to issues. The most common permissions are described below. To learn more about this check out Configuring file attachments.

Jira admin set permissions

- You can attach files and screenshots if your Jira admin has enabled file attachments.
- You need the **Create attachments** permission in the appropriate projects.
- If your Jira admin has disabled thumbnails in Jira's attachment settings, the image files will appear as a list.
- If your Jira admin has disabled ZIP support in Jira's attachment settings, the attachments feature will not be available. You must download the file to your computer before accessing its individual files.
- To remove attachments from an issue, you need one of the following project permissions in that issue's project:
  - **Delete own attachments**—to delete files that you have added to the issue.
  - **Delete all attachments**—to delete files that anyone has added to the issue.

- If you're using Google Chrome, Mozilla Firefox, or Internet Explorer 11, attaching screenshots relies on HTML5 compatibility.

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 you're viewing. You can edit them as 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.

> ⓘ You can only create sub-tasks if your administrator has enabled sub-tasks, and has added the sub-task issue type to the project's issue type scheme.

1. Navigate to the issue you would like to be the parent issue of the sub-task you are about to create.
2. Select **More** > **Create sub-task**. You will see the **Create sub-task** screen.
3. Fill in the details as needed, and then select **Create** at the bottom of the page.

Note that when you create a sub-task, the following values are inherited from the parent task:

- project
- issue security level
- sprint value, if any (only for Jira Software issues)

Tip: You can customize the **Create sub-task** screen to show fields you use most often. To do this, select **Co nfigure fields** at the top right corner of the dialog, and use the **All** and **Custom** links to switch between the default screen and your custom settings. Your changes are saved for future use.

Converting a sub-task to an issue

1. Navigate to the sub-task issue you would like convert.
2. Select **More** > **Convert to i**ssue**.
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.

> (i) 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.

Splitting an issue

Sometimes an issue may actually be such a big chunk of work that it would be better to split it up into several issues. You can easily split your issues into several in your Kanban or Scrum backlog view. You can choose how many issues to split the original into, you can edit the summary if you need to, and if the issue is estimated, and/or in a sprint, these details can also be updated.

When you split the issue, not all fields are copied across:

Copied	System fields	Custom fields
Yes	• Issue type • Project • Description • Environment • Assignee • Reporter • Due Date • Security level • Priority • Labels • Components • Affected versions • Fix Versions	Jira Software • Epic name • Epic link • Epic color • Epic status
No	• Summary • Status • Creator • Created • Updated • Resolution date • Resolution • Votes • Watches • Original estimate • Estimate • Time Spent	Jira Software • Story points • Sprint • Rank Other custom fields

To split an issue:

1. Navigate to the issue you would like to convert in your Kanban or Scrum backlog.
2. Right-click the issue in your backlog and select **Split issue**.
3. Make any adjustments required, you can also add additional issues here by selecting **+ Add another**.
4. Select **Split**.

Once you've split your issue, the relation between the issues is maintained using issue links with the type **Spl it to** and **Split from**.

Restricting access to an issue

When creating (or editing) an issue, you can restrict access to that issue to members of your team who are part of a chosen security level. To be able to set the security level for an issue, your administrator must add you to the appropriate issue security level, and also grant you the 'Set Issue Security' permission for the appropriate projects.

1. Create/edit the relevant issue.
2. In the **Security level** drop-down field, select the desired security level for the issue. You will only see the security levels you belong to.
3. Save the issue. It is now only accessible to members of the specified security level.
   Users who are not members of this security level will not be able to access that issue, or see it in any filters, queries, or statistics.

Creating issues using the CSV importer

If you have the **Create Issue** project permission and the **Bulk Change** global permission for the relevant projects, you can create issues in bulk by using a comma-separated value (CSV) file. To find out what permissions you have, contact your Jira admin.

CSV files are text files that represent tabulated data and are supported by most applications that handle tabulated data, such as Microsoft Excel, Numbers, and databases.

Jira's CSV importer (also CSV file import wizard) allows you to import data from external systems that export their data in a tabulated format. Also, you can create your own CSV file to bulk create or update issues.

> (i) Jira admins have access to more advanced features for importing issues. If you are planning to import a lot of issues from an external application, contact your Jira admin.
>
> Read more about advanced import functionalities for Jira administrators in Migrating from other issue trackers and Importing data from CSV

There are two main steps to use the CSV importer:

1. Preparing your CSV file
2. Running the CSV import wizard

Preparing your CSV file

The CSV importer assumes that your CSV file is based off a default Microsoft Excel-styled CSV file:

- Fields are separated by commas
- Any content that must be treated literally, such as commas, new lines/ (e.g."carriage returns"), or angle brackets, are enclosed in quotes

> (i) For Microsoft Excel and OpenOffice, you don't 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 issue's summary.
- The header row shouldn't contain any punctuation, except for commas separating columns or fields. Otherwise, the import may work incorrectly.

- Commas as column or field separators can't be omitted throughout the file.

```
Summary, Assignee, Reporter, Issue Type, Description, Priority
"Test issue", admin, admin, 1, ,
```

```
Summary, Assignee, Reporter, Issue Type, Description, Priority
"Test issue", admin, admin, 1
```

Encapsulating Jira data structure in your CSV file

To import issues correctly, you should know how to encapsulate different data in your CSV file. In this way, Jira will be able to process them and create or update issues correctly.

Here, you'll find tips on and examples of building a CSV file with multiple-line fields, special characters, multiselect fields, and many more.

Learn more about the specifics of issue fields in Tips for importing CSV data into issue fields.

Capturing data that spans multiple lines

Use double quotation marks (`"`) in your CSV file to capture data that spans multiple lines. For example, during the import, Jira will treat the following as a valid CSV file with a single record:

```
Summary, Description, Status
"Login fails", "This is on
a new line", Open
```

Treating special characters literally

Put the text with special characters in double quotation marks ("") to treat these characters literally. Once imported to Jira, these special characters will be stored as part of Jira's field value. Examples of special characters include carriage returns/enter characters, commas, etc.

If the text contains words or phrases in quotation marks and you want Jira to treat these quotation marks literally, put these words or phrases in another pair of quotation marks.

- Your CSV file may contain a value like: `"Clicking the ""Add"" button results in a page not found error"`. Pay attention that the name of the button is put in two pairs of quotation marks.
- After the import, Jira will store this value as: `Clicking the "Add" button results in a page not found error`. The name of the button is stored in one pair of quotation marks, as it should be.

Aggregating multiple values into single issue fields

You can import multiple values into an issue field that accepts multiple values. For example, **Fix (for) Version**, **Affects Version**, **Component**, **Labels**. To do this, your CSV file must specify the same column name for each value you wish to aggregate into the mapped issue field. The number of column names must match the maximum number of values to be aggregated into the mapped field.

For example:

```
IssueType, Summary, FixVersion, FixVersion, FixVersion, Component, Component
bug, "First issue", v1, , , Component1,
bug, "Second issue", v2, , , Component1, Component2
bug, "Third issue", v1, v2, v3, Component1,
```

In this example, the **Component** field of the second issue and the **Fix Version** field of the third issue will generate multiple values in appropriate issue fields after the import.

> ⚠ Be aware that only a limited number of issue fields support multiple values. The CSV importer will not allow you to import aggregated data into issue fields that only support a single value.

Importing issues into multiple projects

You can import issues from your CSV file into different projects through a CSV file import. To do this:

- The CSV file must have two additional columns with the exact names of **Project Name** and**Project Key**. These are the mandatory values required for the correct import of issues from the CSV file to specific Jira projects.
- Ensure that every issue represented in your CSV file contains the appropriate name and key in these columns for the projects to which they will be imported.

> ⓘ If you aren't a Jira administrator, you won't be able to map the fields **Project Name** and **Proje ct Key** to Jira's fields. You'll need to select different fields for mapping. Learn more about this in Tips for importing CSV data into issue fields.

For example:

```
IssueType, Summary, Project Name, Project Key
bug, "First issue", Sample, SAMP
bug, "Second issue", Sample, SAMP
task, "Third issue", Example, EXAM
```

In this example, the first and second issues will be imported into the "Sample" project (with project key "SAMP") and the third issue will be imported into the "Example" project (with project key "EXAM") , assuming you match the "Project Name" and "Project Key" fields in your CSV file to the **Project name** and **Project key** issue fields respectively during the CSV file import wizard.

Importing work log entries

Your CSV file can contain work log entries. You need to use seconds to track the time spent. For example:

```
Summary,Worklog
Only time spent (one hour),3600
With a date and an author,2012-02-10 12:30:10;wseliga;120
With an additional comment,Testing took me 3 days;2012-02-10 12:30:10;wseliga;259200
```

> ⓘ If you aren't a Jira administrator, you won't be able to map the **Worklog** field to Jira's field. You'll need to select different fields for mapping. Learn more about this in Tips for importing CSV data into issue fields.

Importing to multi select custom fields

Your CSV file can contain multiple entries for the one multi select custom field. Here's an example showing how to populate the multi select custom field with multiple values:

```
Summary,Multi Select,Multi Select,Multi Select
Sample issue,Value 1,Value 2,Value 3
```

Importing cascading choice custom fields

You can import values to a custom field with cascading choice by using the following syntax:

```
Summary, My Cascading Custom Field
Example Summary, Parent Value -> Child Value
```

The `->` separator allows you to import the hierarchy.

> ⓘ Currently Jira does not support importing multi-level cascading select fields via CSV (
> 🔖 **JRASERVER-34202** - Allow CSV import to support Multi-Level Cascading Select plugin fields
> `GATHERING INTEREST`
>).

Running the CSV file import wizard

> ⓘ If your Jira installation has existing data, you should back it up before proceeding with the following steps.

To import a CSV file with your Jira issues:

1. Select **Issues** > **Import Issues from CSV** to open the **Bulk Create Setup** page. If you do not have the option **Import issues from CSV**, your Jira admin must update the Jira Importers plugin to version 6.2.3 or above.

2. On the **Bulk Create Setup** page, select your **CSV Source File**.
    Leave the **Use an existing configuration file** checkbox cleared if you do not have a configuration file or if you want to create a new configuration file. Configuration files specify a mapping between column names in your CSV file's header row and fields in Jira.

> ⓘ 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. It's mandatory for creating or updating an issue.

The **Don't 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 won't see the unmapped fields in issues.

Use this option if:

- You can't 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 don't have a Jira administrator's 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 you've selected the **Map field value** checkbox on the previous page. If you haven't 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, you'll see all values of all fields for which you've 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**.

326

- If you haven't chosen to map values for username-based fields like **Reporter** and **Assignee**, usernames from the file will be automatically mapped to existing usernames (lowercase) in Jira.
- If you import descriptions with line breaks, don't select the **Map field value** checkbox for the **Description** field. Otherwise, Jira may remove line breaks during the import.
- Regardless of whether or not you select the **Map field value** checkbox, Jira will automatically create usernames based on the data in your CSV file if they have not already been defined in Jira.

7. You may want to check your configuration for errors or warnings before running the import. To do it, select **Validate**. If your file contains warnings or errors, we recommend fixing them and uploading the file again. To learn more, download the detailed log of the validation.

ⓘ If you need to import another CSV file with the same (or similar) settings to what you used through this procedure, select the **save the configuration** link to download a CSV configuration file, which you can use at the first step of the CSV file import wizard. For example, you may want to use the same field or value mappings for the next imports.

8. When you're 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 won't be created during the import. Instead, the import wizard will give you a list of any new users that need to be created. You will need to create the users in your external user repository before running the import. • If you have a user-limited license (a personal license), and the number of users is larger than the limit, then the import will be stopped. You'll see the list of users that can't be created. • If Assignee and Reporter aren't mapped, no usernames will be created.
Other fields	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 aren't a Jira administrator, you won't be able to map the following fields in the CSV file to their equivalents in Jira:
>
> • Attachments
> • Comment Body
> • Date Created
> • Date Modified
> • Date Resolved
> • Project Key
> • Project Name
> • Project Type
> • Resolution
> • Status
> • Worklog
>
> To complete the field mapping, select either any other available field or the **Don't 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

Work on your issues 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 enabled file attachments, you and your customers can attach files and screenshots to issues you're working on. See Attaching files and screenshots to issues for more information.

Collaborating on issues

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** (![share icon]) 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 permission for the issue's relevant project. If you do not have this permission, please contact your administrator.

To edit an existing issue, select **Edit** to open the Edit Issue dialog box and modify the issue details. If you want to change the fields you need to edit, select **Configure fields** > **Custom** and choose the fields you want to show or hide. Select **Update** to save your changes.

Commenting on issues

> ⓘ **What permissions do you need?**
>
> To add comments to an issue, 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 won't 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 you've 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 you've 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 can't 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, you'll 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.

Pin a comment

> ⓘ To pin comments on issues, you need to have the **Edit Issue** permission.

Open the issue, Select **Pin** at the bottom of the comment to display it above all other comments.

After you pin a comment, it'll appear in the **Pinned comments** section at the top of the **Comments** tab. The **All comments** section contains all issue comments, including pinned ones.

> ✓ Pinned comments aren't sortable and the recently pinned ones will always appear at the top.

You can unpin, edit, and delete a pinned comment either in the **Pinned comments** or **All comments** section . You can also change the restrictions of pinned comments, making them visible only to particular groups of users.

Link to a comment

Right-click on the comment timestamp, and then copy the link address. Paste the copied permanent link into your email or chat message.

Clicking the permanent link takes you to that particular comment in the Jira issue. If your Jira issue contains an extensive list of comments, the issue page will automatically be scrolled down so that the linked comment is visible.

Restrict a comment

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.

> ℹ️ **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:

You can click a label (e.g. **doc** in the above screenshot) to jump to the Issue Navigator and see a list of all issues that have this label. You can also add the Labels Gadget to your dashboard to quickly find issues with labels relevant to you and your team.

Watching and voting for issues

> ℹ️ **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.

With the granted permissions, you can add other users as watchers to the issue. The user picker in the **Watchers** field will only list users who have access to the project the issue belongs to.

If your administrator has enabled the voting on issues, you can select **Vote for this issue** to encourage the responsible team to resolve or complete the issue.

Reordering sub-tasks on an issue

If you've added sub-tasks to an issue, and need to reorder them, you can drag and drop them on the issue navigator view of the parent issue. If you're using a board in Jira Software, you can also reorder the sub-tasks on the board view. However, these two methods are **independent** of each other. Reordering sub-tasks on the parent issue **will not** reorder tasks on an existing board, and vice-versa.

Linking issues

Issue linking allows you to create an association between two existing issues on either the same or different Jira servers. For example:

- An issue may *relate to* another.
- An issue may *duplicate* another.
- An issue may *block* another.

Issue linking also allows you to:

- Create a new linked issue from an existing issue in a service desk or business project.
- Create an association between an issue and a Confluence page.
- Link an issue to any other web page.

Your Jira administrator can customize the types of links that you can create, see configuring issue linking.

On this page:

- Creating a link to another issue on the same Jira site
- Creating a link to an issue on another Jira site
- Create a new linked issue from an existing issue in a service desk or business project
- Creating a link to a Confluence page
- Creating a link to any web page URL
- Deleting a link
- Searching for linked issues

Issue links within an issue look like this:

Note: Resolved issues (i.e. issues with a Resolution set) are displayed in strike-through font, e.g. ~~DEMO-1~~.

> ⓘ To create links on issues, you need to have the Link Issues permission in the project(s) to which the issues belong.

Creating a link to another issue on the same Jira site

1. Open the issue you wish to link to another issue in the same Jira site.

2. Select **More > Link** to display the **Link** dialog box.

3. Ensure that the **Jira Issue** item is selected at the left of the dialog box and then choose the type of link to be created from the **This issue** drop-down list.
   ℹ️ If your Jira system administrator has configured *fully reciprocal application links* between your Jira site and another one, a **Server** drop-down list may appear above the **This issue** list. If this is the case, ensure your Jira site appears or has been selected from the **Server** list.
4. In the **Issues** field, specify the issue(s) to be linked to your currently viewed/selected issue. There are two ways to do this:
   - Type the full issue key (e.g. **ABC-123**) — or to link to multiple issues, press the 'Enter' key between each typed issue key.
     ℹ️ If you have previously browsed an issue, you can quickly find the issue by typing the first few letters of the issue key (or part of the Summary), which will appear in an 'autocomplete' drop-down list for selection:
     OR:
   - Click the **search for an issue** link to use the **Find Jira issues** popup, which allows you to perform either a simple text search or an advanced search for issues.
5. Optional: Add a **Comment** to describe why you are linking these issues.
6. Click the **Link** button at the bottom of the dialog.

Creating a link to an issue on another Jira site

⚠️ To create this type of link, your Jira system administrator should have configured *fully reciprocal applicatio n links* between your Jira site and the other Jira site containing the issue(s) you want to link to.

1. Open the issue you wish to link to another issue.
2. Select **More > Link** to display the **Link** dialog box.
3. Ensure that the **Jira Issue** item is selected at the left of the dialog box.
   ℹ️ **Note:**
   - This option will not be available if your Jira system administrator has not configured an application link between your Jira site and the remote Jira site.
   - If, after selecting this option, you are prompted for authorization, you may be required to log in to the remote Jira site, which will allow your Jira site to access the remote Jira site *on behalf of your account on the remote Jira site*.
     ℹ️ This behavior means the application links configured between your Jira site and the remote Jira site use OAuth authentication.
4. If your Jira site is connected to multiple remote Jira sites, choose the relevant Jira site from the **Server** drop-down list.

5. Choose the type of link to be created from the **This issue** drop-down list.
6. Type the **Issue** key of the issue on the remote Jira site that you want to link to. Alternatively, you can search for issues on the remote Jira site by clicking the **search for an issue** link, which opens the **Fin d Jira issues** popup.
   ℹ️ You can link to any issue on the remote Jira site to which you have access on that site.
7. Select the **Create reciprocal link** checkbox to create the complementary link on the remote issue you are linking to, back to your issue. For example, if you create a **blocks** link type to a remote issue, the reciprocal link generated on the remote issue will be a **is blocked by** link type back to your local issue.
8. Optional: Add a **Comment** to describe why you are linking these issues.
9. Click the **Link** button at the bottom of the dialog.

Troubleshooting

❌ **Problem:** If you selected the **Create reciprocal link** checkbox, but after clicking the **Link** button, you discover that a reciprocal link from the remote issue back to your issue has not been created, then your Jira system administrator has most likely created only a one-way link from your Jira site to the remote Jira site.

✅ **Solution:** Ask your Jira system administrator to configure *fully reciprocal application links* between your Jira site and the remote Jira site.

❌ **Problem:** If you attempted to create a reciprocal link but received the following message:

'**A reciprocal link from issue 'XYZ-123' back to this issue was not created as the remote Jira server returned the following error: No Link Issue Permission for issue 'XYZ-123'.**' (where 'XYZ-123' is the issue key on the remote Jira site),

then a reciprocal link on the remote Jira site will not have been created, because the user account through which you authenticated on the remote Jira site (at step 3 above) does not have the Link Issues project permission.

✅ **Solution:**

- Ask the Jira project administrator(s) on the remote Jira site to grant your user account the Link Issues project permission for the relevant project(s) to which you need to create issue links.
- Alternatively, if the application link between your Jira site and the remote Jira site use OAuth authentication and you suspect you may have authenticated on the remote site with another user account that does not have the Link Issues project permission, repeat the procedure above but during the authorization step (at step 3), authenticate on the remote site with a user account which has this permission.
  ℹ️ If you are not prompted for authentication during authorization, try clearing your browser's cookies first and repeat the procedure again.

Create a new linked issue from an existing issue in a service desk or business project

> ℹ️ To create a linked issue, you need to have Create issue and Linked Issues permissions in the destination project(s).

To create a linked issue:

1. Open the issue from which you wish to create the linked Jira issue.
2. In the Issue screen, select **More > Create linked issue** to display the **Create Linked Issue** dialog box

The newly created linked issue contains the same Project, Issue Type, and Summary information stored in the original issue. It is also linked to the service desk issue, in this case CTF-2.

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 **Linked issues** field, specify issue(s) to be linked to your new linked issue.
6. Edit the linked issue **Summary.**
7. Edit the **Description** and describe 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* betw een 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.
   ℹ This option is not available if your Jira system administrator has not configured an application link between 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-down list.
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 for authorization, you may be required to log in to the Confluence site, which will allow your Jira site to access the Confluence site *on behalf of your account on the Confluence site*.
  This behavior means the application links configured between your Jira site and the remote Confluence site use OAuth authentication.
    - a. In the first **Search** field, specify one or more search terms that appear in the page you want to link to. This field is mandatory.
    - b. Optional: In the second **Search** field, select the Confluence space to further narrow down the search.
    - c. Click the **Search** button and then the title of the page you want to link to.
6. Optional: Add a **Comment** to describe why you are linking these issues.
7. Click the **Link** button at the bottom of the dialog.

Troubleshooting

❌ **Problem:** If Confluence page links you create show **Failed to load** on the issue or if you attempted to search for a Confluence page but received the following message:

'**Content on the Confluence site could not be accessed because the Confluence server's 'Remote API' feature is disabled. The Confluence system administrator must enable this 'Remote API' feature for Jira to successfully access this content.**'

then Jira was unable to communicate with the Confluence server to either:

- retrieve information about the link or
- conduct a Confluence page search in the **Find a Confluence page** dialog box.

✅ **Solution:**

Ask the Confluence system administrator to enable the **Remote API (XML-RPC & SOAP)** feature, since this Confluence feature is disabled by default. See Enabling the Remote API in the Confluence documentation for details.

Creating a link to any web page URL

1. Open the issue you wish to link to another issue.
2. Select **More > Link** to display the **Link** dialog box.
3. Click the **Web Link** option at the left of the dialog box.
4. Specify the **URL** of the web page you want to link to.
5. Specify the **Link Text** that will appear in the **Issue Links** section of the 'view issue' page and will be hyperlinked to your URL.
6. Optional: Add a **Comment** to describe why you are linking these issues.
7. Click the **Link** button at the bottom of the dialog.

Deleting a link

1. Go to an issue that contains links, and locate the **Issue Links** section.
2. Hover your mouse over the link you wish to delete, and click the **Delete** (trashcan) icon that appears.

Searching for linked issues

You can search for issues that are linked to a particular issue. See Advanced searching for more information.

ⓘ Be aware that this functionality does not extend to issues on a remote Jira server.

Editing multiple issues at the same time

At some point, you may need to change multiple issues at the same time. You can do this by performing a Bulk Change operation.

There are restrictions placed on some of the bulk operations. For example, if you select multiple issues with different workflows, you can only transition them in groups with the same workflow, and one group at a time. You can only bulk change 1000 issues as loading more than that might result in the "Out Of Memory" error. The restrictions are explained further in the relevant sections.

On this page:

- Before you begin
- Delete multiple issues
- Move multiple issues
- Edit multiple issues
- Watch / stop watching multiple issues
- Archive multiple issues

Before you begin

Required permissions

To perform a bulk change operation, you need the appropriate project-specific permission and the global Bulk Change permission. For example, you would need to have both the **Move Issue** and **Bulk Change** permissions to perform the **Bulk Move** operation. If you cannot access the Bulk Change functionality, ask your project admin to grant you the required permissions.

Using the bulk change wizard

The bulk change wizard will progress you through your bulk change. To open the wizard:

- Perform a search with the required filters to produce a list of issues and select **Tools** > **Bulk Change**.
- Bulk edit issues in the Backlog. Use Shift+select to choose multiple issues and then right-click the selected items to start the bulk change operation.

Bulk Operation

● Choose Issues Selected **1** issues from **1** project(s)	**Step 2 of 4: Choose Operation** Choose the operation you wish to perform on the selected **1** issue(s).
● **Choose Operation**	
● Operation Details	
● Confirmation	

○	Edit Issues	Edit field values of issues
○	Move Issues	Move issues to new projects and issue types
○	Transition Issues	Transition issues through workflow
○	Delete Issues	Permanently delete issues from Jira
N/A	Archive Issues	**NOTE**: You might not have the permission to archive one or more issues, or you're trying to archive subtasks instead of issues.
○	Watch Issues	Watch all the selected issues. You will receive notifications when any of these issues are updated.
○	Stop Watching Issues	Stop watching all the selected issues. You will no longer receive notifications when any of these issues are updated.

Next   Cancel

To go back to any step of the operation, select the relevant step in the menu on the left-hand side of the wizard. Selecting **Cancel** will cancel the entire process.

Disable notifications for bulk operations

You can disable mail notifications for a particular bulk operation by deselecting the **Send Notification** checkbox in the bulk operation wizard. For this option to be available, you must be a Jira administrator or project administrator of all the projects associated with your selected issues.

339

> ⓘ **Disable customer notifications.** Some bulk operations, such as **Change comment**, might trigger email notifications to your customers. To prevent a flurry of emails, a Jira admin can temporarily disable outgoing mail in **Administration** (⚙) > **System** > **Mail** > **Outgoing mail**. This setting controls both Jira and customer notifications, so remember to turn it back on when you're done with your bulk edit.

Transition multiple issues

This bulk operation allows you to transition multiple issues through a workflow at the same time. You can only perform one transition bulk operation at a time. You will also need to provide any values required to complete the transition. For example, to close multiple issues, you will need to provide a value for the Resolution field, such as Done, Fixed, or Won't Fix.

To transition multiple issues:

1. Perform a search with the required filters to produce a list of issues.
2. Select **Tools** > **Bulk Change**.
3. Select the issues you'd like to perform the bulk operation on and then select **Next**.
4. Select **Transition Issues** and then select **Next**.
5. Select the available workflow action. The actions available are dependent on the issues (and their associated workflows) that you have selected. Select **Next**.
6. Select a value for any required fields for this transition, and if available, decide whether you'd like to send email notifications. Select **Next**.
7. Review your bulk operation, and select **Confirm** when you are happy with the operation.

Delete multiple issues

This bulk operation allows you to delete multiple issues at the same time. To delete multiple issues:

1. Perform a search with the required filters to produce a list of issues.
2. Select **Tools** > **Bulk Change**.
3. Select the issues you'd like to perform the bulk operation on, and select **Next**.
4. Select **Delete Issues** and then select **Next**.
5. If available, decide whether you'd like to send email notifications. Select **Next**.
6. Review your bulk operation, and select **Confirm** when you are happy with the operation.

Move multiple issues

This bulk operation allows you to move multiple issues at the same time. The issues you're moving need to be mapped to both a project and an issue type, and in doing this, you may need to also map the status and fields of the issues. Subtasks need to be mapped, too.

You can bulk move both standard issues and sub-tasks to another project and issue type, as well as convert a sub-task to an issue and vice versa. To learn more about issue types, see Issue fields and statuses.

You can also select both a sub-task and its parent to bulk move. However, you'll only be asked to move sub-tasks if you're moving a parent issue to another project. This is to maintain the parent/sub-task relationship (i. e. the sub-task is always located in the same project as the parent issue). For example, you have issue B being a sub-task of issue A and you try to bulk move both A and B simultaneously. You will be prompted to select a target project and issue type for issue A. If you select a new project for A, you will be prompted to move the sub-task to a new issue type based on issue A's new project. If you don't change the project for issue A, the sub-task will not be required to be moved.

To move multiple issues:

1. Perform a search with the required filters to produce a list of issues.

2. Select **Tools** > **Bulk Change**.

My open issues | Save as   Share | Export ⌄ | ⚙ Tools ⌄

✓ assignee = currentUser() AND resolution = Unresolved order by updated DESC   ⑦   **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-tasks to the new project. You can also elect to change the issue types of the sub-tasks being moved if you need to.
Select status migration mappings for invalid statuses	As multiple workflows can be active simultaneously, some statuses associated with the collection of selected issues may not be valid in the target workflow. In this case, you should map invalid statuses to valid statuses in your new workflow.
Select values for required fields and fields with invalid values	To adhere to the field configuration scheme associated with the target project and issue type, it may be necessary to update/populate required fields (e.g. fields that are required in the target project, but may not have been in the original project). For each field that needs to be populated, you will be prompted to supply a value. This value will be applied to all issues that are being bulk moved together. 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" ⓘ Versions which have been archived in the target project cannot be selected as the target when performing a bulk move. If you need to move issues into an archived version, you will need to first unarchive the version in the target project. 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 migration mappings for invalid statuses" and "Values for required fields and fields with invalid values" once for each different target project and issue type combination.
When all move parameters — e.g. target project, status mappings and field updates — have been specified for all issues, you will be presented with a confirmation screen displaying all changes that will be made to the issues being moved. The following details are displayed as applicable:

- **Issue Targets:** the target project and issue type
- **Workflow:** the target workflow and invalid status mappings
- **Updated Fields:** new values for fields that require updating
- **Removed Fields:** values to be removed in fields that are not valid in the target

The issues will only be moved once the **Confirm** button is clicked from the confirmation page. If the operation is exited anytime before this step, no changes will be made to the issues

Edit multiple issues

This bulk operation allows you to edit multiple issues at the same time. The bulk edit operations available depend on the issues selected and the nature of the field/s you want to change.

To edit multiple issues:

1. Perform a search with the required filters to produce a list of issues.
2. Select **Tools** > **Bulk Change**.

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

Available Operations	Conditions
Change Affects Version/s	- 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 Tracking fields: 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.

343

To stop watching multiple issues:

1. Perform a search with the required filters to produce a list of issues.
2. Select **Tools** > **Bulk Change**.
3. Select the issues you'd like to perform the bulk operation on, and select **Next**.
4. Select **Stop Watching Issues** and then select **Next**.
5. Review your bulk operation, and select **Confirm** when you are happy with the operation.

Archive multiple issues

> ⓘ This functionality is available with the **Jira Data Center** license.

If you want to archive thousands of issues at once, instead of selecting the issues manually, you can make a bulk change. By default, this option allows you to archive all the issues on the current page or a maximum of 1000 issues. However, a system admin can raise this limit if needed.

The issues will be archived with all their subtasks. Note that issues might have multiple subtasks so archiving might take a while because of a great number of subtasks.

To archive multiple issues:

1. Perform a search with the required filters to produce a list of issues. You can archive all the issues on the current page or a maximum of 1000 issues.
2. Select **Tools** > **Bulk Change**.
3. Select the issues you'd like to perform the bulk operation on, and select **Next**.
4. Select **Archive issues** and then select **Next**.
5. Review your bulk operation, and select **Confirm** when you are happy with the operation.

Moving an issue

Sometimes, an issue may belong to a different project, and you may want to move this issue to another project. You can easily do this by using the **Move issue** wizard.

Before you begin:

- You must have the Move Issues permission for the project that has the issue that you want to move.
- You must have the Create Issues permission for the project that you wish to move your issue to.

If you do not have either of 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

The Move issue wizard allows you to specify another project in your Jira instance to move your selected issue to. As there may be significant differences in the configuration of your original project and target project, the Move issue wizard allows you to change certain attributes of the issue. These include:

- **Issue type** — If your issue is a custom issue type that does not exist in your target project, you must select a new issue type. You can also choose to arbitrarily change the issue type.
- **Issue status** — You may have set up custom issue statuses as part of a workflow. If you have assigned a custom status to your issue, and it does not exist in your target project, you must select a new issue status for your issue. You cannot arbitrarily change the issue status, i.e. the option to change the issue status will only appear if you are required to change it.
- **Custom fields** — If you have defined **required** custom fields for your issue that do not exist in your target project, you must set values for them. You will only be prompted to enter the values for **required custom fields** in the target project that are missing values. If the custom fields of your original project also exist in your target project, and these custom fields are not required in the target project, you may need to set values for them, to move the issue successfully. If you wish to change the existing values for other fields on your issue, you can do this after the move is complete.

To move an issue:

1. View the issue that you wish to move.
2. Select **More** > **Move**.
3. The first page of the Move issue wizard is displayed. Complete the steps required.
4. The confirmation page will display with all of your changes. If you wish to revise any of your changes, you can select the appropriate step in the left-hand menu to return to that page of the wizard. Once you are happy with your changes, select **Move** to move the issue to the target project.
5. Your issue will be moved to the target project and displayed on screen. You can now edit the issue to make further changes, if you wish.

Moving related issues

- If your issue has sub-tasks, the Move issue wizard will also move the sub-tasks to the target project.
- If you're moving an epic, the Move issue wizard will not move the issues in the epic. The epic and the issues in the epic will still be linked to each other, but the issues in the epic will remain in the original project. You will need to move them separately.
- If your issue has values in the Components field, since the components field is project-tied, the target project must have the same component value already in place with the exact same name. Otherwise, the value will be lost during the move operation.

Troubleshooting

- Restricted comments appear to be removed after moving the issue. See this article: Restricted comments disappear after moving an issue to a new project.

Visual editing

Formatting content in Visual mode gives you a What You See Is What You Get (WYSIWYG) experience. Formatting appears as you apply it, and you no longer have to flip to a Preview to see what your content will look like when saved. You still have the option to view the wiki markup by selecting the Text tab. You'll know you have access to the visual editor because you'll see the Visual and Text tabs.

Visual	
Text	

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 Software to track and review, and inform teams about issue 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

The Due Date From and Due Date To fields use a special syntax to denote time period bounds. The syntax uses numbers and abbreviations that follow the numbers to represent what the numbers actually mean. The abbreviations are "w" for weeks, "d" for days, "h" for hours, and "m" for minutes. For example, to specify 10 days in the future, use "10d" or "1w and 3d". To specify a period bound in the past, prefix the value with the "-" sign. For example, to specify 2 days, 4 hours, and 3 minutes ago, use "-2d 4h 3m".

Using advanced search

You can also use Jira Query Language (JQL) to search for issues by due date — see Advanced Searching, and particularly the documentation on the Due field.

Logging work on issues

Jira Software provides the flexibility to set your estimation and tracking statistics differently, depending on your team's needs. Time Tracking features project schedule planning and time expectations management. Thanks to its reports, you can see the original and current time estimates for all the issues, and whether they are ahead or behind the original schedule. Product teams often need to be able to estimate how long a product will take to deliver from sprint to sprint; as they work through the stories, the team will develop a cadence of completing <x> units of work they had estimated. This is also called velocity. The team has to track the amount of estimation units they have completed from sprint to sprint to know how much they can fit into each future sprint, and have conviction that all the work will be completed. You will be able to easily monitor the tracking time info by just watching the issue time tracking color bars.

On this page:

- Before you begin
- Setting a time estimate for an issue
- Logging work on an issue
- Editing a work log entry
- Deleting a work log entry
- Customized Jira installations

Here's how time tracking appears on an issue:

- The Estimated field displays the amount of time originally anticipated to resolve the issue
- The Remaining field displays the amount of time currently anticipated to resolve the issue
- The Logged field displays the amount of time logged working on the issue so far
- Choosing to include sub-tasks displays the aggregated time of an issue and all its sub-tasks

When you log time for the first time, the time spent is subtracted from the original estimate, and the resulting value is automatically presented in the remaining estimate. When subsequent work is logged, any time spent is subtracted from the remaining estimate.

Before you begin

- Make sure your Jira administrator has enabled the Time Tracking feature.
- Make sure you have the Work on Issues, Delete Work Logs, and Edit Work Logs project permissions.

> (i) Note that anyone with the Browse Project permission can view time tracking information on an issue.

Setting a time estimate for an issue

Teams can set a time estimate for an issue in order to calculate how long it will take to solve the issue.

1. Open the issue and select **Edit**.
2. Scroll down the Edit issue window to fill in the following time tracking fields:

Field	Description
Original Estimate	Amount of time you believe is required to solve the issue. If you want to change original estimate values once they have logged work time, ask your Jira administrator to disable legacy mode on time tracking.
Remaining Estimate	Amount of time you believe is required to solve the issue in its current state.

If the Jira time tracking feature is in legacy mode, you will only see the original estimate field if work has not been logged. Once work time has been logged, you will only see the remaining estimate field.

> ✅ **Tips:**
> - You can specify additional time units after a time value 'X', such as Xw, Xd, Xh, or Xm, to represent weeks (w), days (d), hours (h), and minutes (m), respectively. If you type a number without specifying a time unit (e.g. if you type '2' instead of '2h'), the default time unit that your Jira administrator specified will apply.
> - Default conversion rates are 1w = 5d and 1d = 8h.

3. Select **Update**.

When work is first logged against the issue, the **Time Spent** is subtracted from the **Original Estimate**, and the resulting value is automatically presented in the **Remaining Estimate**. When subsequent work is logged, any **Time Spent** is subtracted from the **Remaining Estimate.**
Additionally, once work has been logged on an issue, various reports based on the time tracking information become available.

Logging work on an issue

Once you have started to work on a specific issue, you can log your work by following these steps:

1. Select the issue you want to log time on.
2. Go to **More > Log Work**.
3. Fill in the following **Log Work** fields, and select **Log**:

Log Work field	Description
Time spent	The amount of time spent on the issue. This is the aggregate amount of time that has been logged against this issue.
Date started	Date and time when you started this unit of work.
Remaining 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.
	Comments are copied to the **Workflow Description** by default, but your Jira administrator can change this option in the 'Copy Comment to Workflow Descriptions' settings. If this setting is disabled:
	- The work log entry may be visible to anyone. If this is a concern, you need to edit this work log entry after creating it to modify its visibility. - You have to manually copy comments to a workflow description once you have logged work.

You can also log work while resolving or closing an issue by closing it and editing the log work fields. Select the padlock icon to set the work logged to be viewable only by members of a particular project role or group.

Editing a work log entry

You can edit your own work log entries if you have been granted the Edit Own Work Logs permission. You can also edit other people's work log entries if you have been granted the Edit All Work Logs permission.

Deleting a work log entry

You can delete your own work log entries if you have been granted the Delete Own Work Logs permission You can also delete other people's work log entries if you have been granted the Delete All Work Logs permi ssion.

1. Go to the desired issue, and open the **Work Log** tab.
2. Hover over the work log entry to display the actions for the entry on the right side.
3. Select the entry you want to delete, and click the trash can icon. You will be prompted to choose how the Remaining Estimate is affected by deleting the work log:

Option field	Description
Auto adjust	Choose this option to automatically add the time spent value to the current remaining estimate value.
Leave existing estimate	Select this option if you do not want to change the issue remaining estimate value.
Set estimated time remaining	Choose this option to manually set the issue's remaining estimate value to the specified amount.
Increase estimated time remaining	Select this option to increase the estimated remaining.

4. Click **Delete**.

Customized Jira installations

Jira applications can be customized by your Jira administrator by adding the Log Work and Time Tracking fields to the customized screens. This way, you can log work and specify time estimates on the same Jira screen when performing any Jira operation, such as editing, creating an issue, or transitioning an issue to another status.

If you want to work *and/or* specify time estimates on the same Jira screen:

1. Navigate to the issue and view its details.
2. Perform the customized Jira operation that allows you to log work *and* specify time estimates on the same Jira screen. For example, assuming that your Jira administrator has added the **Time Tracking** fi elds to the **Resolve Issue Screen,** and assuming this screen also retains the default **Log Work** fields, select **Workflow > Resolve Issue** at the top of the issue.

- If your Jira administrator has configured the Log Work fields as optional, then you can choose whether or not to log work by checking the Log Work checkbox.
- If your Jira administrator has made logging work mandatory, you will not see the Log Work checkbox, and will instead need to log work when transitioning an issue.

Flagging an issue

You can flag an issue to indicate that it's important. The card of a flagged issue is displayed in yellow in the Scrum backlog, Active sprints of a Scrum board, Kanban backlog (if enabled), and Kanban board, with the 'flag' icon replacing the priority icon.

Flagging or unflagging an issue

1. Navigate to your desired board.
2. Click either **Scrum backlog**, **Active sprints**, **Kanban backlog** (if enabled), or **Kanban board**.
3. Click the issue that you want to flag or unflag.
4. In the Issue Detail View, select **Add flag** or **Remove flag** from the 'cog' drop-down. You can also right-click on the issue > **Add flag** or **Remove flag**.

You can also add a comment when you're adding a flag to or removing a flag from an issue. You may want to do this to indicate your reason for adding or removing the flag.

In the Issue Detail View, select **Add flag and comment** or **Remove flag and add comment** from the 'cog' drop-down. You can also right-click on the issue > **Add flag and comment** or **Remove flag and add comment**. After adding your comment, it will appear in the Comments section of the issue, with an indication that a flag was added or removed accordingly.

Searching for flagged issues

The flag for an issue is stored in a custom checkbox field named "Flagged", which has only one value: Impediment.

This means that you can use this JQL query to find flagged issues: `Flagged = Impediment`

Ranking an issue

Rank your issues to organize the tasks in a sprint more effectively. By ranking issues, you actually arrange issues according to their relative importance or urgency. For example, you have two issues that are of 'High' priority. With Jira Software ranking, you can choose which of these two issues have a higher priority or ranking than the other.

When you rank an issue, you also change its relative priority in its current column and swimlane.

To rank an issue:

1. Navigate to your desired board.
2. Navigate to your **Scrum backlog**, **Active sprints**, **Kanban backlog** (if enabled), or **Kanban board**.
3. Rank an issue by dragging and dropping it to a higher or lower position in the Scrum or Kanban backlog, or to a column in the Active sprints or Kanban board.

Screenshot: dragging and dropping an issue in the Backlog

Screenshot: dragging and dropping an issue to a column in the Active sprints

Tips and notes:

- You can use Keyboard Shortcuts **'s' + 'b'** to move an issue to the bottom of its current column, or **'s' + 't'** to move it to the top.
- You can send multiple issues to the top or bottom of the backlog. Select the issues while pressing and holding down the 'Shift' or 'Ctrl' key, and then right-click > **Top of Backlog** or **Bottom of Backlog**.
- When you rank an issue that has sub-tasks, all of the sub-tasks are automatically moved with the issue.
- Sub-task issues can only be ranked in relation to their 'sibling' issues.

ⓘ Please note:

- You can only rank issues if ranking has been enabled — see Enabling ranking.
- You can only rank issues if you have the 'Schedule Issue' and 'Edit Issue' permissions for the issue you want to move higher or lower on your board.

Related pages

- Sorting by rank does not show expected values
- Enabling ranking
- Using your Scrum backlog
- Using Active sprints
- Using your Kanban backlog
- Monitoring work in a Kanban project

Referencing issues in your development work

If your administrator has connected Jira Software to your development tools, you can easily synchronize your development work with your issues — just reference an issue key(s) in your commits, branches, pull requests, etc, and you will enable the following:

- Show links to your development work on the issue in a **Development** panel
- Show details of the development work in the **Development** column on your list view of your search
- Automatically transition the issue to a new status, if your administrator has set up **workflow triggers**
- Collate your issues and development work on the **Version details** page, which is used to track and release the version

Note, when you reference your issue key(s), your connected development tools will also have links back to the relevant issues.

For more details on how to connect a development tool with Jira Software, seeIntegrating with development tools

The following table describes how to reference an issue key in a commit, branch, pull request, or review. In all cases, **the issue key must conform to the default Jira key format** – that is, two or more uppercase letters (`[A -Z][A-Z]+`), followed by a hyphen and the issue number. For example, ABC-123.

Event	Dev Tool	Instructions
Create commit	Bitbu cket, GitHu b, GitHu b Enter prise, Fishe ye	Include the issue key in the commit message. For example, a commit message like this "TIS-1 Initial commit" will automatically transition the TIS-1 issue from 'To Do' to 'In Progress'.
Create branch	Bitbu cket, GitHu b, GitHu b Enter prise, Fishe ye	Include the issue key in the branch name when you create the branch. For example, if you name your branch "TIS-2 feature", it will automatically transition the TIS-2 issue from 'To Do' to 'In Progress'.
Create /Reopen /Decline Merge pull request	Bitbu cket, GitHu b, GitHu b Enter prise	Do at least one of the following: - Include a commit in the pull request that has the issue key in the commit message. Note, the commit cannot be a merge commit. - Include the issue key in the pull request title. - Ensure that the source branch name includes the issue key. For example, if you create a pull request that has "TIS-3" in the title, it will automatically transition the "TIS-3" issue from 'In Progress' to 'In Review'. If you reopen, decline, or merge the pull request, it will also transition the "TIS-3" issue accordingly.

Start/Reject /Abandon /Close review	Cruci ble	Include the issue key in the review title when you create the review. For example, if you name your review "TIS-4 New story" and start the review, it will automatically transition the TIS-4 issue from 'In Progress' to 'In Review'. If you reject, abandon, or close the review, it will also transition the "TIS-4" issue accordingly.
Create build plan	Bam boo	A build is automatically linked to an issue if one of the build's commits includes the issue key in its commit message. The issue key must be included in the commit to activate this feature.
Create deployment project	Bam boo	A deployment to an environment, such as Production or testing, is linked to an issue if a commit associated with the deployment contains the issue key in its commit message. The issue key must be included in the commit to activate this feature.

If you reference a Jira issue in the development tool, the information from this tool will display in the **Developme nt** panel in the issue view. Learn more about the Development panel in Viewing the development information for an issue.

Estimating an issue

Before you begin

Estimating stories in your backlog helps you predict how long it would take
you to deliver certain portions of the backlog. Note that this discussion
refers to the best practices we've implemented as the main path in Jira
Software — you can choose not to use this approach if you feel it's really
not suitable for your team.

ℹ This page only applies to Scrum boards.

Estimate an issue

Before a sprint starts, you need to enter the Original Estimates of your issues. And as you work on the
issues during the sprint, you may need to adjust the Remaining Estimates as necessary.

To enter the Original Estimate, do the following for each issue:

1. Navigate to the **Backlog** of your desired board.
2. Click the issue that you want to set the Original Estimate for.
3. In the Issue Detail View, type your estimate in the **Estimate** field.

> ℹ The type of units used by the 'Estimate' field (e.g. hours) is affected by your Estimation Statistic —
> see Configuring estimation and tracking.

To adjust the Remaining Estimate:

1. Navigate to the **Active sprints** of your desired board.
2. Click the issue that you want to adjust the Remaining Estimate for.

3. In the Issue Detail View, type your estimate in the **Remaining** field.

ℹ️ The type of units used by the 'Remaining' field (e.g. hours) is affected by your Tracking Statistic — see Configuring estimation and tracking.

Concepts about estimation

Here are some concepts to consider when estimating issues in Jira Software.

In Scrum, there is a distinction between estimation and tracking. *Estimation* is typically performed against Primary Backlog Items (PBIs, usually stories), and is used to work out how long portions of the backlog might take to be delivered. *Tracking* refers to monitoring the progress of a sprint, to be sure that all stories included in the sprint will be delivered.

Tracking is often performed by breaking down stories into tasks, and applying hour estimates to them during sprint planning, then monitoring the remaining time in a burndown during the sprint.
In traditional development environments, estimation is done this way:

1. A team estimates items in 'man-hours' – and these estimates are assumed to be accurate.
2. The team then calculates the total number of man-hours for the backlog of a project.
3. The team then divides the total number of man-hours by the number of people on the team, and the man-hours in a week. This becomes the forecast date for the project.

These estimates are often inaccurate because they don't consider the following:

- The natural estimation characteristics of the team – meaning, over- and under-estimation are not considered
- Unexpected interruptions during the man-hours allocated to the items
- The performance of the team members themselves over time

When the estimates become inaccurate, the team then exerts time and effort in trying to 'force' the estimates to be accurate. This makes the man-hour approach difficult — if not impossible.
In the Scrum world, teams don't try to achieve estimation accuracy. Instead, they aim to achieve 'reliable velocity'.

Velocity is a measure of the number of estimation units that a team tends to complete from sprint to sprint. After their first few sprints, most teams will achieve a reasonably consistent velocity. Armed with velocity and estimates on the PBIs in the backlog, teams can predict more accurately how long portions of the backlog will take to complete.

The key is, the estimation unit doesn't matter – as long as it becomes reasonably predictable from sprint to sprint. For example, teams can use 'ideal hour' estimates, but it's neither necessary or expected that those hours will have any relationship to elapsed time. If a team has a 'man-hour' capacity of 120h in each sprint but a velocity of 60h, that makes no difference because you can still use the 60h velocity to estimate the number of sprints that portions of the backlog will take to complete — and therefore, the elapsed time.

Many people then start wondering where 'the other 60 hours' went, thereby implying that there is something wrong with team productivity. But that's usually got nothing to do with it: a team's estimates merely represent their view of how hard items will be, taking into consideration the team's natural behavior (such as over- and under-estimation), as well as organizational overhead, etc. The velocity is all that matters from a planning perspective.

Since the units are not related to time, most teams now choose to use story points as their estimation unit. A story point is an arbitrary number that measures the complexity of one story relative to others. In effect, story points clearly break the mental link with time.

For a team's velocity to reach a stable state, the team must estimate each backlog item with the same level of accuracy. At the risk of repeating the obvious, the goal of velocity is to be able to look at a backlog of not particularly well-understood stories, and understand how many sprints it will take to complete. This requires a similar level of uncertainty for all of the estimates in the backlog.

There is a counter-intuitive implication here — that teams should estimate each item once, and not change that estimate even if they discover new information about the item that makes them feel their Original Estimate was wrong. If the team were to go ahead and update estimates, this 'discovery of new information' will happen regularly. This leads to the backlog having some items that have higher accuracy, but most that don't. This would pollute velocity because sprints with a larger percentage of high accuracy estimates will complete a different number of units compared to those with a lower percentage of high accuracy estimates. As a result, the velocity could not be used for its primary purpose — that is, for estimating the number of sprints it will take for a team to complete a set of not-well-understood stories in the backlog. Therefore, it's critical to use the first estimates so that the team's velocity realistically represents their ability to *complete* a certain number of units of not-well-understood work far ahead into the future.

Consider the following scenario:

- Issue X has an Original Estimate of 5 days.
- Before the next sprint is planned, the team realizes that the Original Estimate was too optimistic, and that the issue actually takes 15 days.

Some people would argue that using the Original Estimate will endanger the sprint's success, because the team will take in what they think is 5 days of work into the next sprint when it's actually 15 days of work.

However, the inaccurate estimate of 5 days is unlikely to be an isolated occurrence. In fact, the estimates are always going to be wrong (some very little, some wildly so). This will often be discovered after the sprint has started rather than before. As long as the team estimates the same way across the whole backlog, this will work itself out over time. For example, if they always underestimate, they may find that for a 10-day sprint with 4 team members, they can only really commit to 20 days of their estimation unit. If they have established a stable velocity, then this has no effect. From a planning perspective, we can still reliably estimate how much work we'll get done in upcoming Sprints.

When the team is about to start a sprint, they can use the velocity as an indication of items from the backlog that they can realistically complete. The velocity here is based on the number of items they have successfully completed in the past. However, some people may question how this can be right when the Original Estimates won't include information about work that may have already been done, or information about how hard a particular item of work is.

As an example, consider the following scenario:

- An issue has an Original Estimate of 10 days.
- The team works 5 days on the issue in the current sprint.
- The team discovers a bad bug somewhere else in the project, and they decide that fixing that bug in the current sprint is far more important than completing issue X as planned.
- The sprint gets finished, and the issue returns to the backlog.

In the next sprint, the team would be tempted to update the estimate for the issue to 5 days, and use that to make their decision whether or not to include it in the sprint. The implication is that they might not include enough work in the next sprint if they used the issue's Original Estimate of 10 days. However, the reason that the task was not completed previously is because of unplanned work — and it's unrealistic to assume that this won't happen again in the future, perhaps even in the next sprint. Thus, the 10-day estimate is a realistic number to use in the absence of certainty. As a result, the cost of the unplanned work that may happen is eventually accounted for in the Original Estimate. Even if the work does turn out to be insufficient for the next sprint, the team will correct that by dragging more work into the sprint.

In the same example, consider if this were the only issue in that sprint and will be the only issue in the next. If the issue is completed in the second sprint, and we use the Remaining Estimate, then the velocity will be (0d + 5d) / 2 = 2.5d. However, the team can clearly complete more work than that in future sprints. If we use the Original Estimate, then the velocity will be (0d + 10d) / 2 = 5d. The use of the Original Estimate accounts for the fact that the team cannot commit to 10d in every sprint because unplanned work will likely make that impossible. It also realistically accounts for the fact that unplanned work will not happen in every sprint. Many teams break down stories into sub-tasks shortly before the sprint begins so they can use the stories for tracking. This raises the possibility of using the sum of the estimates on the sub-tasks as a way to decide which issues to commit to in the sprint (and potentially for velocity).

As described above, tracking is really a separate process from estimation and velocity. The estimates that are applied to sub-tasks clearly have higher accuracy than those that were originally applied to the story. Using them for velocity would cause the velocity to have both high and low accuracy estimates, making it unusable for looking further out in the backlog where stories have only low accuracy estimates.

In addition, only items at the top of the backlog are likely to have been broken into tasks. Using task estimates for velocity means that the velocity value could only predict the time to complete the backlog up to the last story that has been broken into tasks.

Lastly, using sub-task roll-up to decide sprint commitment is risky because, unlike velocity value, it doesn't consider the overhead of unplanned work and interruptions.
More and more industry leaders are moving away from hour estimates, and are now using the story point approach. This makes sense because in a sprint, the main questions to be answered are:

- How much work can we realistically commit to completing this sprint?
- How long will this part of the backlog take to deliver?

The story point approach based on original estimates can deliver the answers to these questions without the anxiety around 'accuracy' that teams feel when asked to estimate in hours.

The Jira Software team itself uses the approach described in this article, and has established a reliable velocity that we use to plan work months in advance — even when new work has been encountered during those months. We recommend this approach because while it is sometimes counter-intuitive, it is also powerful, fast, and simple.

All of that said, one of the key precepts of agile is finding the way that works for you. So Jira Software does support the alternatives described above, including the use of remaining estimates for sprint commitment, hours for estimation, and hour estimates on sub-tasks.

Transitioning an issue

Before you begin

Before you can transition issues to columns on a board, your JIRA administrator or board administrator will have already mapped workflow stat uses to your board's columns. To know more about configuring these mappings, see Configuring columns . Also, you can only transition an issue if you have project permission to move that issue to the status in the JIRA workflow to which the column is mapped.

Jira Software will respect your Jira workflow configuration, and will trigger all your JIRA workflow validations, conditions, functions, and notifications when you transition issues (See Workflows for more information.) However, note that boards currently don't auto-assign issues to you.

To transition an issue to a different JIRA workflow status on a board:

1. Navigate to your desired board.
2. In the **Active sprints** *(Scrum board)* or **Kanban board** *(Kanban board)*, transition an issue by dragging and dropping the issue from one column to another.
   When you start dragging an issue, you can only transition the issue to columns whose backgrounds turn blue. When the issue is hovered over the target column, the blue background turns green.
3. If the target column is associated with more than one status, when you hover the issue over that column, each status will appear as a separate section with a dashed outline. Complete the transition to the appropriate status by dropping the issue onto the section of the column.
4. If a dialog box opens for your workflow transition status, you must complete all mandatory fields (indicated by a red asterisk) before submitting the form.

If an issue has sub-tasks, and all sub-tasks have been completed, you need to resolve the issue itself. When you move the last sub-task to the 'Done' column, Jira Software will prompt you to move the parent issue as well. If you resolved the sub-tasks in native Jira Software instead, a button will be displayed on the parent issue the next time you visit the Active sprints page.

Depending on the following conditions, you can transition multiple issues from the backlog to the next status in your project's workflow:

- You have the transition issues permission.
- The statuses of the issues are already mapped to the board's columns.
- Your project's workflow supports the particular transition for the multiple issues selected, and all transition conditions are met.
- The issues you're transitioning have the same set of transitions.
- The issues you're transitioning don't have any associated transition screen.
- You can select multiple issues by using the **Shift** or **Ctrl** key while clicking the issues. However, you can't transition multiple issues by dragging them to another column. You can only move multiple issues within the same column.

Viewing the development information for an issue

If your administrator has connected Jira Software to a compatible development tool, a **Development** panel will be displayed on the View Issue screen, and when you view a search for issues in a list view, you can also view a **Development** column. Depending on which tools Jira Software is connected to, the Development panel and Development column provide the following functionalities:

- Bitbucket Cloud or Bitbucket Server: view and create branches, view commits, and view and create pull requests
- Fisheye/Crucible *(Git/Subversion/Perforce/CVS)*: view branches (not create), view commits, and view and create reviews
- Bamboo: view the status of builds and deployments
- GitHub or GitHub Enterprise: view branches (not create), view commits, and view and create pull requests

Before you begin

> ⓘ Your administrator should set up Jira Software and connect development tools to it correctly. Refer your administrator to Integrating with development tools.
>
> You must have the View development tools permission to see the Development panel.

Make your development information available in Jira Software

You must reference the issue key in your commit, branch, pull request, etc if you want it to display as a link in the Development panel or column. For information on how to reference issue keys correctly, see Referencing issues in your development work.

View the Development panel

If everything has been set up correctly, you will see a Development panel on each of your issues, similar to the screenshot below. The Development panel provides you with just enough information to evaluate the status of an issue's development at a glance.

1. **Development panel**

Notes:

The pull request(s) status in the Development panel is:

| OPEN | if there is at least one open pull request.

| MERGED | if there are no open pull requests, and at least one pull request has been merged.

| DECLINED | if there are no open or merged pull requests, and at least one pull request has been declined.
The review(s) status in the Development panel is:

| REVIEW | if there is at least one review in 'Review' status.' (yellow with black writing)

| APPROVAL | if there are no reviews in 'Review' status, and at least one review is in 'Approval' status. (black)

| SUMMARIZE | if there are no reviews in 'Review' or 'Approval' statuses, and at least one review is in 'Summarize' status. (black)

| REJECTED | if there are no reviews in 'Review', 'Approval', or 'Summarize' statuses, and at least one review is in 'Rejected' status. (red)

| APPROVAL | all reviews are in 'Closed' status.

Note, 'Draft' and 'Abandoned' reviews are not shown.
The build(s) status in the Development panel is:

✅ if all the different builds (for example, unit tests, functional tests, deploy to staging) have passed.

🚫 if at least one run failed for any build by any linked build server.

View the Development column

If everything has been set up correctly, you will be able to view a Development column for searched issues in a list view, similar to the screenshot below. The Development column provides you with just enough information to evaluate the status of an issue's development at a glance, if you click on an icon or link, you'll see all the information from the linked development tool.

View Development tool information on a card

You can view Development tool information on a card by navigating to your board, selecting **Board > Configure > Card layout**, and adding the **Development** field. You'll then be able to see information from your connect Development tools on your cards, giving you more context when viewing issues on a board.

Investigate and action the information

If you want to investigate something in the Development panel or column, you can click the item to display a dialog with more details. The dialog provides links for you to action or drill into. Note, you may be prompted to authenticate with the linked application first.

For example, say that the Development panel was showing this: 3 builds ⊕. You could click 3 builds to open a dialog showing which of the three builds are failing. If you wanted to investigate further, then you could click the build plan or build result to view it in Bamboo.

Here are the actions you can do via the Development panel:

BITBUCKET

Click **Create branch** in the Development panel to open your connected SCM, and start the process for creating a branch. If you have multiple applications connected, then you can choose where you'd like to create the branch. The key for the Jira issue will be automatically added to the name of the branch.

BITBUCKET	**FISHEYE**	**GITHUB**

Click ***n* branches** in the Development panel to open a dialog in Jira Software that shows the branches for the linked SCM. If Jira Software has been linked to more than one SCM, a tab will show for each SCM product (e.g. Bitbucket). The branches will be grouped under each SCM in these tabs.

TIS-45: 2 branches

FishEye / Crucible 1   Stash 1

Repository	Branch	Pull request	Action
Apollo UI	feature/TIS-45-email-non-registered-users-to-sign	MERGED	Create pull request

- Click a repository or branch to open the linked SCM at the relevant repository/branch.
- Hover over a **Pull request** status (e.g.   MERGED   ) to show a popup displaying a link to the pull request.
- Click **Create pull request** to create a pull request for the branch (to merge it back to master).

BITBUCKET   FISHEYE   GITHUB

Click *n* **commits** in the Development panel to open a dialog in Jira Software that shows the commits and related files for the linked SCM. If Jira Software has been linked to more than one SCM, a tab will show for each SCM product (e.g. Bitbucket). The commits will be grouped under each SCM in these tabs. If a commit is greyed out, it has been merged through a pull request.

TIS-73: 2 unique commits (and 2 duplicates)

FishEye / Crucible 2   Stash 2

Apollo UI (Stash)   Show all files

Author	Commit	Message	Date	Files
	3047fd41d44	TIS-73 Fix the highlighting	22/Aug/14	1 file
	9d5d94c1e26	TIS-73 License update and minor fix	22/Aug/14	2 files

- Click a repository or commit to open the SCM at the relevant repository/branch.
- If Jira Software is linked to more than one SCM, the dialog may show duplicate commits across tabs. For example, you may have pointed FishEye and Bitbucket to the same repository.
- *(Bitbucket only)* If the commits belong to a fork of a repository, they will be grouped under the fork. The fork will also have a link to the original repository, "**Fork of <repository name>**".

BITBUCKET   GITHUB

See the 'See repository branches' section above.

BITBUCKET   GITHUB

Click *n* **pull requests** in the Development panel to open a dialog on Jira Software that shows the pull requests for the linked SCM(s). If Jira Software has been linked to more than one SCM, a tab will show for each SCM product (e.g. Bitbucket). The pull requests will be grouped under each SCM in these tabs.

TIS-57: 1 pull request

ID	Title	Status	Author	Reviewer	Updated
#1	Bugfix/TIS-57 buttons need to be red	OPEN			12/Aug/14

- Click a pull request to open it in the linked SCM.
- Hover over a user icon to show the user's name.

CRUCIBLE

Click *n* **reviews** in the Development panel to open a dialog in Jira Software that shows the reviews.

- Click a review to open it in Crucible.

BAMBOO

Click *n* **builds** in the Development panel to open a dialog in Jira Software that shows the builds.

- Click a plan or build result to view it in Bamboo.

BAMBOO

Click **Deployed** *(to environment)* in the Development panel to open a dialog in Jira Software that shows the deployment.

- Click an environment or release to view it in Bamboo.

Viewing agile information for an issue

The **Agile** panel displays additional agile information about the issue you are currently viewing and — depending on the type of project the issue is part of — lets you find boards that display the viewed issue or the sprints that the issue belongs to.

Here's an example of the content of the **Agile** panel displayed for an issue in a Scrum project:

1. The parent project's active sprint that the issue belongs to and its end date
2. The parent project's completed sprints that the issue belongs to and their end dates
3. The option to find agile boards that display the currently viewed issue

For issues in Kanban projects, the **Agile** panel only displays the option to find other agile boards that display the currently viewed issue.

Finding related agile boards

Besides displaying information about the active or completed sprints that an issue belongs to, the **Agile** panel lets you search for agile boards displaying the viewed issue or a sprint that the issue belongs to.

To make sure that Jira remains responsive and the results are relevant to you, when you run a search for board displaying the viewed issue, the search scope is limited to:

- boards that explicitly mention an issue's project key
- your recently viewed boards
- a maximum of 100 boards

> ⓘ Since we don't search all the boards that you may have access to, the lack of search results doesn't mean that there don't exist any boards displaying an issue. If you need to broaden the search scope, use Advanced search.

At the same time, the limits described above don't apply to searching for Scrum boards displaying an active or completed sprint. When you run a search for boards displaying specific sprints, Jira will include all of the boards that you're allowed to access in the scope of the search.

Here's how to find the agile boards displaying an issue or sprint:

- To find the agile boards that display the issue you're viewing, select **Find on a board**, and then select a search result to jump to the respective board view.
- To find Scrum boards that display a sprint listed in the **Agile** panel, select the sprint name, and then select one of the search results to jump to the respective board view.

> ✓ If you don't want to interrupt your workflow, you can also run the search in a new browser tab.

Printing issue cards

Whether you're planning work or working on issues for Scrum or Kanban projects, it may also be good to print out these issue cards. You and your team can use the printed cards on a physical board, which can replicate your board on Jira Software. If you want to, you can print a single issue card or multiple issue cards. You can also print all issue cards in your current board.

The printed issue cards include the following issue details:

- Summary
- Issue type
- Issue key
- Issue priority
- Estimate
- Assignee
- Epic (optional in Active sprints/Kanban board)
- Version (when printing from the backlog)
- Up to 3 extra fields, depending on your card layout configuration

The printed issue cards fit on A4-, A3-, or Letter-sized pages in both portrait and landscape modes.To print an issue card:

1. Navigate to your desired board.
2. Navigate to your **Scrum backlog**, **Active sprints**, **Kanban backlog** (if enabled), or **Kanban board**.
3. Right-click the issue that you want to print, and then select **Print selected card**. The Print window will appear.
4. From the **Card size** menu on the print preview page, select the card size for printing.
5. Select **Print** button and configure printing options in your web browser.

> ⊘
>
> - You can print multiple issues from a column on the current board. To do this, hold down the **S hift** key and select the cards you want to print. Then, right-click one of the selected cards and select **Print selected cards**.
> - To print all issues in the backlog or on the current board, in the upper-right corner of the screen select **Board** > **Print cards**.
> - When printing from a Scrum backlog, you can filter the cards for printing by selecting the backlog or a sprint from the **Select** menu on the print preview page.
> - If you want to print only a subset of the issues in the backlog or on the current board, configur e a quick filter to control which issues should appear.

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

- See Customizing cards for more information on configuring issue cards in Jira Software.
- See Using your Kanban backlog and Configuring columns for more information on using and enabling the Kanban backlog.

Processing issues with Smart Commits

When you manage your project's repositories in Bitbucket or GitHub, or use Fisheye to browse and search your repositories, you can process your Jira Software issues using special commands, called Smart Commits, in your commit messages.

You can:

- comment on issues
- record time tracking information against issues
- transition issues to any status defined in the Jira Software project's workflow

There are other commands available if you use Crucible for software reviews. See Using Smart Commits in the Crucible documentation.

A single Smart Commit command cannot span more than one line (i.e. you cannot use carriage returns in the commit message), but you can add multiple commands to the same line, or multiple commands on separate lines. See this example below.

Smart Commit commands

The basic command line syntax for a Smart Commit message is:

```
<ignored text> <ISSUE_KEY> <ignored text> #<COMMAND> <optional
COMMAND_ARGUMENTS>
```

Any text between the issue key and the Smart Commit command is ignored.

There are three Smart Commit commands you can use in your commit messages:

- comment
- time
- transition

Comment

Description	Adds a comment to a Jira Software issue.
Syntax	`<ignored text> ISSUE_KEY <ignored text> #comment <comment_string>`
Example	`JRA-34 #comment corrected indent issue`
Notes	- The committer's email address must match the email address of a single Jira Software user with permission to comment on issues in that particular project.

Time

Description	Records time tracking information against an issue.
Syntax	`<ignored text> ISSUE_KEY <ignored text> #time <value>w <value>d <value>h <value>m <comment_string>`
Example	`JRA-34 #time 1w 2d 4h 30m Total work logged`

Notes	This example records 1 week, 2 days, 4 hours and 30 minutes against the issue, and adds the comment 'Total work logged' in the **Work Log** tab of the issue. • Each value for w, d, h and m can be a decimal number. • The committer's email address must match the email address of a single Jira Software user with permission to log work on an issue. • Your system administrator must have enabled time tracking on your Jira Software instance.

Workflow transitions

Description	Transitions a Jira Software issue to a particular workflow state.
Syntax	`<ignored text> ISSUE_KEY <ignored text> #<transition_name> <comment_string>`
Example	`JRA-090 #close Fixed this today`
Notes	This example executes the close issue workflow transition for the issue and adds the comment 'Fixed this today' to the issue. Note that the comment is added automatically without needing to use the #comment command. You can see the custom commands available for use with Smart Commits by visiting the Jira Software issue and seeing its available workflow transitions: 1. Open an issue in the project. 2. Click **View Workflow** (near the issue's **Status**). The Smart Commit only considers the part of a transition name before the first space. So, for a transition name such as `finish work`, then specifying `#finish` is sufficient. You must use hyphens to replace spaces when ambiguity can arise over transition names, for example: `#finish-work`. If a workflow has two valid transitions, such as: • `Start Progress` • `Start Review` A Smart Commit with the action `#start` is ambiguous because it could mean either of the two transitions. To specify one of these two transitions, fully qualify the transition you want by using either `#start-review` or `#start-progress`. • When you resolve an issue with the `#resolve` command, you cannot set the **Resolution** field with Smart Commits. • If you want to add a comment during the transition, the transition must have a screen associated with it. • The committer's email address must match the email address of a single Jira Software user with the appropriate project permissions to transition issues.

Advanced examples

Multiple commands over multiple lines on a single issue

Syntax	`<ISSUE_KEY> #<COMMAND_1> <optional COMMAND_1_ARGUMENTS> #<COMMAND_2> <optional COMMAND_2_ARGUMENTS> ... #<COMMAND_n> <optional COMMAND_n_ARGUMENTS>`

Commit message	```JRA-123 #comment Imagine that this is a really, and I mean really, long comment #time 2d 5h```
Result	Adds the comment 'This is a really, and I' (but drops the rest of the comment) and logs 2 days and 5 hours of work against issue JRA-123.

Multiple commands on a single issue

Syntax	```<ISSUE_KEY> #<COMMAND_1> <optional COMMAND_1_ARGUMENTS> #<COMMAND_2> <optional COMMAND_2_ARGUMENTS> ... #<COMMAND_n> <optional COMMAND_n_ARGUMENTS>```
Commit message	```JRA-123 #time 2d 5h #comment Task completed ahead of schedule #resolve```
Result	Logs 2 days and 5 hours of work against issue JRA-123, adds the comment 'Task completed ahead of schedule', and resolves the issue.

A single command on multiple issues

Syntax	```<ISSUE_KEY1> <ISSUE_KEY2> <ISSUE_KEY3> #<COMMAND> <optional COMMAND_ARGUMENTS> etc```
Commit message	```JRA-123 JRA-234 JRA-345 #resolve```
Result	Resolves issues JRA-123, JRA-234 and JRA-345. Multiple issue keys must be separated by whitespace or commas.

Multiple commands on multiple issues

Syntax	```<ISSUE_KEY1> <ISSUE_KEY2> ... <ISSUE_KEYn> #<COMMAND_1> <optional COMMAND_1_ARGUMENTS> #<COMMAND_2> <optional COMMAND_2_ARGUMENTS> ... #<COMMAND_n> <optional COMMAND_n_ARGUMENTS>```
Commit message	```JRA-123 JRA-234 JRA-345 #resolve #time 2d 5h #comment Task completed ahead of schedule```
Result	Logs 2 days and 5 hours of work against issues JRA-123, JRA-234 and JRA-345, adds the comment 'Task completed ahead of schedule' to all three issues, and resolves all three issues. Multiple issue keys must be separated by whitespace or commas.

Get Smart Commits working

It's easy to get Smart Commits working for your instance of Jira Software:

Tool	**Connection instructions**

Bitbucket Server	Create an application link between Jira Software and Bitbucket Server. See Linking Bitbucket Server to Jira. Then, enable smart commits in Jira Software: 1. Log in to Jira as a user with Jira administrator permissions. 2. In the upper-right corner of the screen, select **Administration** ⚙ > **Applications.** 3. Under **Integrations**, select **Application links**. 4. Next to the application you want to configure, select **Actions** > **Smart commits**. 5. Select the checkbox for the account you want to enable smart commits for.
FishEye	Create an application link between Jira Software and Fisheye. See Linking to another application. Then, enable Smart Commits in Jira Software: 1. Log in to Jira as a user with Jira administrator permissions. 2. In the upper-right corner of the screen, select **Administration** ⚙ > **Applications.** 3. Under **Integrations**, select **Application links**. 4. Next to the application you want to configure, select **Actions** > **Smart commits**. 5. Select the checkbox for the account you want to enable smart commits for.
Crucible	Create an application link between Jira Software and Crucible. See Linking to another application. Then, enable Smart Commits in Jira Software: 1. Log in to Jira as a user with Jira administrator permissions. 2. In the upper-right corner of the screen, select **Administration** ⚙ > **Applications.** 3. Under **Integrations**, select **Application links**. 4. Next to the application you want to configure, select **Actions** > **Smart commits**. 5. Select the checkbox for the account you want to enable smart commits for.
Bitbucket Cloud, GitHub	First, link your Jira Software and Bitbucket or GitHub accounts. See Connect Bitbucket Cloud to Jira Software Server and Linking GitHub accounts for details. **Check if Smart Commits are enabled for new repositories:** 1. Log in to Jira as a user with Jira administrator permissions. 2. In the upper-right corner of the screen, select **Administration** ⚙ > **Applications**. 3. Under **Integrations**, select **accounts**. 4. Click the settings icon ⚙ for the account. 5. Select **Enable Smart Commits on new repositories**. **Enable or disable Smart Commits on individual repositories:** 1. Log in to Jira as a user with Jira administrator permissions. 2. In the upper-right corner of the screen, select **Administration** ⚙ > **Applications**. 3. Under **Integrations**, select **accounts**. 4. Select (or deselect) the **Smart commits** option for a repository.

Notes

- Smart Commits only support the default Jira Software issue key format. This format is two or more **uppercase** letters, followed by a hyphen and the issue number. For example, JRA-123.

- A DVCS such as Git includes a user's email address in the commit data. Users configure this email address in their local system. Smart commits functionality requires that this email address matches **ex actly one** email address in the Jira Software user base. If the email address matches multiple users in Jira Software, or the user does not have permissions for the requested action, the Smart commit action will fail. The commit itself will still succeed, and will show on the issue. Mismatched email addresses is a common reason why Smart commits fail to work as expected. If a Smart commit fails, Ji ra Software sends an email notification to either the Jira Software user or to the DVCS user (if a Jira Software user can't be identified). In rare cases, Jira Software doesn't have either of these email addresses, and the Smart commit fails silently.
- Smart commits relies on the Jira DVCS Connector Plugin when your repositories are hosted in Bitbucket Cloud or GitHub. The plugin is bundled with Jira Software, but if necessary, a Jira administrator can install it directly from within the Jira administration area. To do this, go to **Manag e apps** > **Find new apps**. See Installing Marketplace apps.

Managing your user profile

You can manage your Jira settings (e.g. your password, email address, or the format in which you would like to receive email notifications) in your user profile. Your user profile also displays recent work in the Activity Stream, and contains useful shortcuts to issues you have been working on or reported.

To manage your user profile:

Choose **your user name** at top right of the screen, then choose **Profile**.

On this page:

- [Editing your user details](#)
- [Changing your avatar](#)
- [Choosing your homepage](#)
- [Managing email notifications](#)
- [Managing your user preferences](#)
- [Managing service desk preferences](#)
- [Managing your OAuth and login tokens](#)

Editing your user details

> ⓘ If your instance is using an external user management system like Crowd, these options may not be available to you.

In the **Details** section on the **Summary** page, click the 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 **My Jira Home** section:
   - Dashboard
   - Issue Navigator
   - Boards (available if you're using Jira Software)

   ℹ️ Your page will be reloaded the Jira home page you selected.
3. *(Optional)* To verify that your Jira home page has been reset, log out and log back in to Jira again. You should be taken directly to the Jira home page you selected in the previous step.

Managing email notifications

In the **Preferences** section on the **Summary** page, click the **edit** icon [pencil icon] at the top-right of the section to open the **Updated User Preferences** dialog box. You can then manage the following:

- Change the **Email Type** to change the format (plain text or HTML) in which Jira sends its outgoing email notifications.
- In **My Changes**, Choose between making Jira send you email notifications about issue updates made by either both you and other people (**Notify me**) or other people only (i.e. **Do not notify me**).

Managing your user preferences

The global defaults for most of the user preferences below can be set by your Jira administrator; however, you can override these default settings by changing the following:

- The **Page Size**, or number of issues displayed on each Issue Navigator page
- Your preferred **language** from the drop-down list. If you don't see your preferred language in the list, see Translating Jira for more information.
- Your **time zone** specified in your profile doesn't match the time zone of the computer you are working on, Jira will ask if you want to update this selected time zone setting. All time fields in Jira will now be displayed in your preferred time zone.
- Choose the **Sharing** setting for when you create new filters and dashboards, which can be either shared with all other users (**Shared**) or restricted (**Unshared**).
- Choose to enable or disable Jira's keyboard shortcuts feature.
- Choose between allowing Jira to make you an **autowatcher** of any issue that you create or comment on.
- Choose how you want your external links to open. They can open in the existing or a new tab.

Managing service desk preferences

Service desk agents can enable or disable the **Pre-populated commenting** field by editing their user profiles. This setting can help save time by pre-filling conversation greeting text when agents comment on customer issues. When enabled, the text **Hi <Reporter_name>,** and – **<Agent_name>** appears in the comment field and in the email notification sent to customers.

Managing your OAuth and login tokens

An OAuth access token is issued by Jira to give gadgets access to restricted data on an external, OAuth-compliant web application or website (also known as a "consumer"). Check out Allowing OAuth access for recommendations on when to issue or revoke OAuth access tokens.

If you are accessing your Jira applications in a public environment, you can clear 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 Jira data that you can access.

Before you begin

Your Jira administrator must link your Jira instance and the consumer using an application link and OAuth. For example, if you want to add a Jira gadget to your Bamboo homepage, then your Jira administrator must first approve Bamboo as an OAuth consumer.

> ⓘ If you want to integrate with OAuth 2.0, see Integrating with OAuth 2.0.

Issuing OAuth access tokens

To allow a gadget to access the same Jira data that you can, Jira issues it an OAuth access token. The OAuth token is unique to the gadget.

1. When you use a Jira gadget on a consumer (such as Bamboo) and this gadget requires access to your Jira data, 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 Jira data, click **Allow**. The gadget can access your Jira data until you revoke the token.
4. To view tokens you have issued, go to your **Profile** > **Tools** > **View OAuth Access Tokens**:

Authorized Applications			
The following applications are using your account to access JIRA data			
Bamboo on server-gdn-bamboo.internal.atlassian.com	Approved on Apr 26, 2016 at 7:09 AM	READ AND WRITE ACCESS	Revoke Access
Confluence	Approved on Apr 22, 2016 at 5:53 AM	READ AND WRITE ACCESS	Revoke Access
JDOG - JIRA Team Dogfood on jdog.jira-dev.com	Approved on Apr 27, 2016 at 12:43 AM	READ AND WRITE ACCESS	Revoke Access
Stash on stash.atlassian.com	Approved on May 12, 2016 at 6:10 AM	READ AND WRITE ACCESS	Revoke Access
Atlassian JIRA Extranet - Special Projects on extranet.atlassian.com	Approved on Jul 15, 2016 at 12:01 AM	READ AND WRITE ACCESS	Revoke Access
Atlassian JIRA on jira.atlassian.com	Approved on May 30, 2016 at 1:06 AM	READ AND WRITE ACCESS	Revoke Access

Revoking OAuth access tokens

You can revoke an OAuth access token to deny a Jira gadget access to your Jira data. When you revoke access, the gadget can only access public data on your Jira instance.

1. To view tokens you have issued, go to your **Profile** > **Tools** > **View OAuth Access Tokens**
2. Next to the application whose OAuth access you wish to revoke, click **Revoke Access**.
3. You may be prompted to confirm this action. If so, click **OK**.
4. The gadget's access token is revoked and the Jira gadget can only access public Jira data.

Requesting apps

The Atlassian Marketplace website offers hundreds of apps that administrators can install to enhance and extend your Jira applications. If the app request feature is enabled for your instance, you can submit requests for Marketplace apps directly to your administrator.

The 'Atlassian Marketplace for Jira' page presents an integrated view of the Marketplace website from within the Jira user interface. The page offers the same features as the Marketplace website, such as app search and category filtering, but tailors the browsing experience to Jira application users.

This in-product view of the Marketplace gives day-to-day users of the Atlassian applications, not just administrators, an easy way to discover the apps that can help them work. When you find an app of interest, you can submit a request with just a few clicks.

Submitting an app request

1. From anywhere in the application, open your profile menu and choose **Atlassian Marketplace**.
2. In the Atlassian Marketplace page, use the search box to find apps or use the category menus to browse or filter by apps by type, popularity, price or other criteria. You can see what your fellow users have requested by choosing the **Most Requested** filter.
3. When you find an app that interests you, click **Request** to generate a request for your administrator.
4. Optionally, type a personal message to your administrators in the text box. This message is visible to administrators in the details view for the app
5. When ready, click **Submit Request**.
6. Click **Close** to dismiss the 'Success!' message dialog box.

At this point, a notification appears in the interface your administrators use to administer apps. Also your request message will appear in the app details view, visible from the administrator's 'Find New apps' page. From there, your administrator can purchase the app, try it out or dismiss requests.

Updating an app request

After submitting the request, you can update your message at any time. Click the **Update Request** button next to the listing in the 'Atlassian Marketplace' page to modify the message to your administrator.

The administrator is not notified of the update. However, your updated message will appear, as you have modified it in the details view for the app immediately.

Using keyboard shortcuts

Keyboard shortcuts are a great way for you to speed up editing, navigating, and for performing actions without having to take your fingers off the keyboard.

Some keyboard shortcuts require additional permissions or applications, and depend on how your Jira administrator(s) have configured permissions for your user account and which applications are installed.

On this page:

- View keyboard shortcuts
- Enabling and disabling keyboard shortcuts

View keyboard shortcuts

- Choose the question mark icon () at top right of the screen, then choose **Keyboard shortcuts**.
- When viewing a page, press **Shift + /** .

The Keyboard Shortcuts dialog is displayed and shows commands for the operating system and browser that you are using. The dialog is divided into sections for the following information:

- **Global shortcuts** - shortcuts that can be used when you are in any part of Jira
- **Navigating issues** - shortcuts for navigating through issues
- **Issue actions** - shortcuts for working with issues
- **App specific** - any application-specific shortcuts. These shortcuts only work in the listed application.

If you have other Jira applications installed, you may have additional keyboard shortcuts available. For example, if you have Jira Software installed, you will see a series of additional keyboard shortcuts in the lower-right of this dialog box (and some additional **Global** keyboard shortcuts specific to Jira Software in the upper-left section). However, the keyboard shortcuts in the **Agile Shortcuts** section only function in Jira Software, and not in a Jira context.

Enabling and disabling keyboard shortcuts

Keyboard shortcuts are enabled by default. However, you can disable them on a per-user basis in the Keyboard Shortcuts dialog box.

1. Ensure you are logged in and open the Keyboard Shortcuts dialog box (see above).
2. At the bottom of the Keyboard Shortcuts dialog box, click **Disable Keyboard Shortcuts** or **Enable Keyboard Shortcuts**.
   You can also disable or re-enable keyboard shortcuts by editing the Preferences section of your user profile. See Managing your user profile for more information.

Modifier keys

Some keyboard shortcuts require modifier keys to be pressed simultaneously, along with a single 'action' key. Modifier keys may differ, depending on your combination of operating system and web browser. The following table identifies the modifier keys for some supported web browsers and operating systems:

Web Browser	Mac OS X	Windows	Linux /Solaris	Notes
Firefox	Ctrl	Alt + Shift	Alt + Shift	In Firefox, it is possible to customize 'Modifier key shortcuts'. Please read Mozilla's documentation for more information.
Internet Explorer		Alt		Typing a 'Modifier key shortcut' that leads to a link requires you to press the 'Enter' to complete the action.
Safari	Ctrl + Alt /Option	Ctrl		

Chrome	Ctrl + Alt /Option	Alt + Shift	Alt + Shift	

Reporting

Reporting is an activity that you will be doing throughout a project. Jira Software has a range of reports that you can use to show information about your project, versions, epics, sprints, and issues.

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

Search the topics in 'Reporting':

Generating a report

1. Navigate to the desired board and click **Reports**. The last report viewed will be displayed.
2. Click **Switch report** to view a different report. The reports in this list are specific to agile development. See the 'Reports for Scrum projects' or 'Reports for Kanban projects' sections below for more details.
3. If you want to view reports that are not specific to agile development, select **All reports** from the **Switch report** drop-down, and view the reports that are not in the 'Agile' section. See the 'General reports for analyzing issues' section below for more details.

Reports for Scrum projects

Chart	Applies to	Purpose
Burndown Chart	Sprints	Tracks the total work remaining, and projects the likelihood of achieving the sprint goal. This helps your team manage its progress and respond accordingly.
Burnup Chart	Sprints	Tracks the changes in sprint scope and progress and helps visualize a sprint's completed work compared with its total scope. This helps your team understand what the progress is at each point of the release process or sprint.
Sprint Report	Sprints	Shows the work completed or pushed back to the backlog in each sprint. This helps you determine if your team is overcommitting or if there is scope creep.
Control Chart	Projects, versions, or sprints	Shows the cycle time for your product, version, or sprint. This helps you identify whether data from the current process can be used to determine future performance.
Cumulative Flow Diagram	Any period of time	Shows the statuses of issues over time. This helps you identify potential bottlenecks that need to be investigated.
Epic Report	Epics	Shows the progress towards completing an epic over time. This helps you manage your team's progress by tracking the remaining incomplete and unestimated work.
Epic Burndown	Epics	Similar to the Epic Report, but optimized for Scrum teams that work in sprints. Tracks the projected number of sprints required to complete the epic. This helps you monitor whether the epic will release on time, so you can take action if work is falling behind.
Release Burndown	Versions	Similar to the Version Report, but optimized for Scrum teams that work in sprints. Tracks the projected release date for a version. This helps you monitor whether the version will release on time, so you can take action if work is falling behind.

Velocity Chart	Sprints	Tracks the amount of work completed from sprint to sprint. This helps you determine your team's velocity, and estimate the work your team can realistically achieve in future sprints.
Version Report	Versions	Tracks the projected release date for a version. This helps you monitor whether the version will release on time, so you can take action if work is falling behind.

Reports for Kanban projects

Chart	Applies to	Purpose
Control Chart	Projects, versions, or sprints	Shows the cycle time for your product, version, or sprint. This helps you identify whether data from the current process can be used to determine future performance.
Cumulative Flow Diagram	Any period of time	Shows the statuses of issues over time. This helps you identify potential bottlenecks that need to be investigated.

General reports for analyzing issues

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 Tracking Report *	Shows time tracking information on issues for a particular version of a project.
	
	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 **Estimate d 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:
	
	• 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.
	
	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.
	
	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.
	
	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 if sub-tasks are 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 has enabled time tracking.

Reports available in Confluence

If you have connected Jira 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, Jira SQL 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.

Burnup Chart

The Burnup Chart provides a visual representation of a sprint's completed work compared with its total scope. It offers insights on your project's progress, as well as offers warnings to help you maintain your project's health; you can instantly identify problems such as scope creep or a deviation from the planned project path.

- Before you begin
- Viewing your chart
- Understanding the Burnup Chart

Before you begin

- The Burnup Chart only applies to Scrum boards.
- Story Points on sub-tasks are not included in the Burnup Chart. (Only Story Points on *parent tasks* are included.)

Viewing your chart

1. Click **Projects** then select the relevant project.
2. Click **Reports**, then select **Burnup Chart**.
   - To choose a different sprint, or a different measurement for the vertical axis, simply click the drop-down menus.

You can view the Burnup Chart for individual sprints in your project, meaning you can see the amount of work completed vs the amount of work remaining on a day-to-day basis.

Understanding the Burnup Chart

- The vertical axis represents the amount of work, and can be measured in different ways such as story points, issue count, or estimates. The horizontal axis represents time in days.
- The chart shows the red work scope line and the green completed work line (completed stories, tasks, fixed incidents for example) in relation to the grey Guideline, which is a theoretical line showing the daily completion necessary to meet the deadline.The distance between the lines on the chart is the amount of work remaining. When the project has been completed, the lines will meet.

- The spikes in work scope mean that something has been added to the milestone while spikes in completed work mean a story has been completed. This information can be crucial at retrospective to understand if, for example, too much work has ben added or the work has been more complex than estimated.
- Examine the 'Work scope' line to identify any scope creep.
- You can examine the actual and planned start and end dates for each sprint.

389

Burndown Chart

Screenshot: Burndown Chart

A Burndown Chart shows the actual and estimated amount of work to be done in a sprint. The horizontal x-axis in a Burndown Chart indicates time, and the vertical y-axis indicates cards (issues).

Use a Burndown Chart to track the total work remaining, and to project the likelihood of achieving the sprint goal. By tracking the remaining work throughout the iteration, a team can manage its progress, and respond to trends accordingly. For example, if the Burndown Chart shows that the team may not likely reach the sprint goal, then the team can take the necessary actions to stay on track.

On this page:

- Before you begin
- Viewing the Burndown Chart
- Understanding the Burndown Chart
- Next steps

Before you begin

- The Burndown Chart only applies to Scrum boards.

Viewing the Burndown Chart

1. Navigate to your desired board.
2. Click **Reports**, then select **Burndown Chart**.
    - To choose a different sprint, click the sprint drop-down.
    - To choose a different estimate statistic, click the estimation statistic drop-down. This change will be saved for you, for when you next visit this chart.

*Tip: Click **How to read this chart** at the top of the report to view a short description of the report.*

Understanding the Burndown Chart

Before you start using the Burndown Chart, you should get to know how it works. The following information will help you understand the key functionalities of the Burndown Chart:

- The Burndown Chart is board-specific – that is, it will only include issues that match your board's saved filter.
- The vertical axis represents the estimation statistic that you have configured for your board.
- The Burndown Chart is based on your board's column mapping. An issue is considered to be 'To Do' when it is in a status that has been mapped to the left-most column of your board. Similarly, an issue is considered to be 'Done' when it is in a status that has been mapped to the right-most column of your board. See Configuring columns for more information.
- If the grey 'Guideline' line does not show, the sprint may have been started before any issues were assigned to it, as described in this KB article.

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Gadgets for JIRA applications
- Configuring working days
- Configuring estimation and tracking

Control Chart

The Control Chart shows the cycle time and lead time for your product, version, or sprint. It takes the time spent by each issue in a particular status and maps it over a specified period of time. The chart also shows the average, rolling average, and standard deviation for this data.

A Control Chart helps you identify whether data from the current sprint can be used to determine future performance. The less variance in the cycle time of an issue, the higher the confidence in using the mean (or median) as an indication of future performance.

Here are some of the ways that you could use a Control Chart:

- analyze your team's past performance in a retrospective
- measure the effect of a process change on your team's productivity
- provide external stakeholders with visibility of your team's performance
- for Kanban, use past performance to set targets for your team

1. **Issue details:** See the cycle time for an issue.
2. **Time scale:** Select a time period to display on the control chart.
3. **Zoom in:** Highlight an area of the chart to focus on a specific time period.
4. **Refine report:** Select the columns and filters you want data for.

Viewing the Control Chart

1. Navigate to your desired board.
2. Click **Reports**, then select **Control Chart**.
3. Configure the chart as desired. The screenshot at the top of this page highlights the controls that you can use to configure the Control Chart.

*Tip: Click **How to read this chart** at the top of the report to view a short description of the report.*

⚠ If you're using Internet Explorer 8, the Control Chart won't work.

Printing the Control Chart

To print the report, view the report and use the print functionality for your browser. The report will fit on either A4- or Letter-sized pages in both portrait and landscape modes (note, there is a known issue printing in landscape using Chrome).

Understanding the Control Chart

Before you start using the Control Chart, you should learn how it works. The following questions and answers cover the key functionalities of the Control Chart

Cycle time is the time spent *working* on an issue. Typically, it's the time from when the work on an issue begins to when this work is completed. It also includes any other time spent working on the issue. For example, if an issue is reopened, worked on, and completed again, the time for this extra work is added to the cycle time.

Lead time is similar to cycle time but is the time from when an issue is *logged* (not when work begins) until the work is completed.
Cycle time

The statuses for calculating cycle time depend on the workflow you're using for your project. You should configure the Control Chart to include the statuses that represent the time spent working on an issue. Note that the Control Chart will select these statuses automatically.

For example, if you're using the **Jira Software Development** workflow, the work on an issue may start when its status changes to **In progress** and finish when the issue transitions from **In review** to **Done**.

You can set this on the Control Chart by selecting **In Progress** and **In Review** as the **columns** so that you'll see the time that issues have spent in these two statuses.

The average cycle time is calculated by adding the time that each issue has spent in each of the selected statuses and dividing the total by the number of issues.

For example, you have five issues that have spent the following amount of time in the statuses **In Progress** and **In Review**.

Issue	In Progress	In Review
ISSUE-1	1 day	1 day
ISSUE-2	2 days	2 days
ISSUE-3	3 days	1 day
ISSUE-4	1 day	3 days
ISSUE-5	2 days	3 days

The average cycle time is (1+1+2+2+3+1+1+3+2+3)/5 = 3.8 days.
You can also configure the Control Chart to show lead time instead of cycle time.

Just like for cycle time, the statuses for calculating lead time depend on the workflow you're using for your project. For example, if you're using the **Jira Software Development** workflow, an issue gets raised when it acquires the status **To Do** and gets completed when it transitions from **In Review** to **Done**.

You should configure the Control Chart to include the statuses that represent the time from when an issue was raised until when it was done. Note that the Control Chart will select these statuses automatically.

To set this on the Control Chart, select **To Do** and **In Review** as the **columns** so that you'll see the time that issues have spent from when they were raised to when they were completed.

The average lead time is calculated by adding the time that each issue has spent in each of the selected statuses and dividing the total by the number of issues.

For example, you have five issues that have spent the following amount of time in the statuses **To Do**, **In Progress**, and **In Review**.

Issue	To Do	In Progress	In Review
ISSUE-1	2 days	1 day	1 day
ISSUE-2	1 day	2 days	2 days
ISSUE-3	2 days	3 days	1 day
ISSUE-4	4 days	1 day	3 days
ISSUE-5	3 days	2 days	3 days

The average cycle time is (2+1+1+1+2+2+3+3+1+4+1+3+3+2+3)/5 = 6.4 days.
The rolling average (blue line on the chart) is issue-based, not time-based. For every issue shown on the chart, the rolling average (at that point in time) is calculated by taking the issue itself, X issues before the issue and X issues after the issue, then averaging their cycle times. 20% of the total issues displayed (always an odd number and a minimum of 5 issues) is used in the calculation.

For example, in the screenshot below, at the point of time where an issue (green dot) is shown, the rolling average is calculated as follows:

1. Take the issue plus four issues before and four issues after (nine issues total).
2. Average the cycle times for the nine issues.
3. Map the blue line to the calculated average.

If the **Timeframe** is reduced to 'Past two weeks', the number of issues used would reduce, as there are fewer total issues available to use for the calculations.

This method produces a steady rolling average line that shows outliers better (i.e. rolling average doesn't deviate as sharply towards outliers). The rolling average line is also easy to understand, as the inflections are related to the positions of issues.

ℹ️ If you would like to know more about why the rolling average calculation is based on a percentage of the total issues, rather than a time period, see Comparing different methods of calculating the rolling average on the Control Chart.
The blue shaded area of the control chart represents the standard deviation — that is, the amount of variation of the actual data from the rolling average.

The standard deviation gives you an indication of the level of confidence that you can have in the data. For example, if there is a narrow blue band (low standard deviation), you can be confident that the cycle time of future issues will be close to the rolling average.

As shown on the chart legend, each dot represents an issue or a group (cluster) of issues:

- The vertical placement of the dot represents the cycle time for the issue, i.e. the 'Elapsed Time'. For a cluster of issues, the dot is placed at the average cycle time for the issues.
- The horizontal placement indicates when the issue(s) transitioned out of the last status selected on the chart (in **Columns**). For example, if you are using the 'Jira Software Development' workflow and have selected 'In Progress' and 'In Review' as the columns on the Control Chart, the dots will indicate when the issue transitioned out of the 'In Review' status.

If the maximum **Elapsed Time** value on the chart is less than 30 days, then a linear scale is used for the y-axis. If it is 30 days or greater, than a cube-root power scale is used.

When you change the **Timeframe**, you may include issues with an elapsed time of greater than 30 days when you previously did not, or vice versa. This will change the scale, as described above.

Linear scale for Elapsed Time

Cube-root power scale for Elapsed Time

Tips and examples

Learn how to tweak your Control Chart to show the data you need with the following examples:

Tip 1: Remove unwanted outliers

The Control Chart can help you identify outliers. On closer examination, you may determine that certain outliers are invalid due to human error. For example, you may have a story that was started but stopped, then eventually dropped back to the backlog, but not returned to the 'To Do' status. The time that the issue spent 'In Progress' would incorrectly skew the data for your Control Chart.

To remove unwanted outliers from your Control Chart, add a label to each outlier issue (e.g. outlier) and create a Quick Filter with this JQL: `labels is EMPTY or labels not in (outlier)`. Configure your Control Chart to use this Quick Filter.

Example Control Chart with invalid outliers

Example Control Chart with invalid outliers removed (note the smaller scale for 'Elapsed Time')

Tip 2: Remove triage casualties

In a Control Chart, you generally want to track the issues that are resolved as 'Fixed'. Issues that are triaged and resolved as a duplicate, answered, tracked elsewhere, etc can skew the data, bringing the average cycle time down considerably.

To remove triage casualties from your Control Chart, create a Quick Filter with this JQL: `resolution in (Fixed)`. Configure your Control Chart to use this Quick Filter.

Example Control Chart including issues where the resolution is not 'Fixed'

Example Control Chart excluding issues where the resolution is not 'Fixed' (note the higher average cycle time)

Tip 3: Exclude current work

The Control Chart shows data for issues that have been in a selected column, but are no longer in a selected column. This gives the cycle time (total elapsed time) for the issues. However, by default, this will include issues that are still moving across the board.

To view the data for completed work only in your Control Chart, create a Quick Filter with this JQL: `status in (Resolved, Closed)`. Configure your Control Chart to use this Quick Filter.

Example Control Chart including all issues

Example Control Chart including issues where the status is 'Resolved' or 'Closed' only

Learn how to interpret a Control Chart with the following examples:

Example 1:

- The productivity of the team is increasing: indicated by the downward trend of the rolling average.
- The cycle time of future issues are likely to be close to the rolling average (2 days or less): indicated by the low standard deviation (narrow blue shaded area).

Example 2:

- The team's productivity is pretty consistent: indicated by rolling average being close to the average.
- There are obvious outliers on Tue 14 and Wed 15 (7 days and 10 days elapsed time respectively, compared 2 days on average) that should be investigated.
- Data is becoming more predictable over time.

Known issues

If you encounter an issue that is not on this list, please raise it in our issue tracker.

JSWSERVER-16449 - Converting days to weeks in Control Chart disregards working days configuration
GATHERING IMPACT

JSWSERVER-20516 - The "tick" box in the names of the Columns and Swimlane section of the Control chart is overlapping CLOSED

🔴 ~~**JSWSERVER-16393**~~ - The "Control Chart" report displays variables for duration, rather than text

`CLOSED`

🔴 ~~**JSWSERVER-21120**~~ - After install Jira SM the Control Chart report receives Uncaught TypeError

`CLOSED`

🔴 **JSWSERVER-12122** - Using recursive transitions impact negatively Agile reports (Control Chart, Cumulative flow diagram) `GATHERING IMPACT`

🔴 **JSWSERVER-21491** - When there is no mapped status, accessing control chart report throws error

`GATHERING IMPACT`

Next steps

ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Configuring working days
- Configuring Quick Filters

Comparing different methods of calculating the rolling average on the Control Chart

The new Control Chart uses a different calculation for the rolling average, than the old Control Chart. On this page, we compare different methods for calculating the rolling average, and show how the issue-based calculation (Example 3) is better.

Consider how the rolling average (blue line) is mapped in the examples shown below. All three examples use the same data:

- Example 1: Window based on time, centered on each day
- Example 2: Window based on time, centered on each issue
- Example 3: Window based on a number of the total issues, centered on each issue

Example 1: Window based on time, centered on each day

In this example, the rolling average is calculated and mapped for **each day** on the chart. This calculation is used in the old Control Chart.

On each day, the average is calculated by doing the following:

- Determine a **window of time** (e.g. two days), based on the overall timeframe of the chart.
- Use the issues in the window to calculate the average cycle time.

One-week timeframe (one-day window)   **Two-week timeframe (two-day window)**

This approach has a number of problems. In these charts, you can see that the last issue is clearly an outlier. However, as it is too far apart from other issues, the rolling average line simply jumps up, particularly in the one-week timeframe. The rolling average line also bends at unexpected places on the chart.

The one-week timeframe also shows no standard deviation (light blue shaded area) around the outlier, although it is a bit better when zoomed out to a two-week timeframe. However, the two-week timeframe shows other problems — the rolling average rises and even fluctuates before the very first issue.

Example 2: Window based on time, centered on each issue

In this example, the rolling average is calculated and mapped for **each issue** on the chart.

At each issue, the average is calculated by doing the following:

- Determine a **window of time** (e.g. two days), based on the overall timeframe of the chart.
- Use the issues in the window to calculate the average cycle time.

One-week timeframe (one-day window)   **Two-week timeframe (two-day window)**

This chart is the same as the one on the left, but is zoomed out to two weeks, i.e. includes an additional week prior to the original.

This approach is better than the approach in Example 1. By centering around the issues points, the variation in the moving average line is easier to understand — each completed issue causes a change. The moving average also starts at the first point, instead of moving up from zero.

However, note that the outlier is still not handled well. The rolling average still jumps up and shows no standard deviation.

Example 3: Window based on a number of the total issues, centered on each issue

In this example, the rolling average is calculated and mapped for **each issue** on the chart. This calculation is used in the new Control Chart and is described in more detail in Control Chart.

At each issue, the average is calculated by doing the following:

- Take a **set number of issues** (e.g. five issues), based on the total number of issues in the chart.
- Use the issues in the window to calculate the average cycle time.

One-week timeframe

Two-week timeframe

This approach improves upon both Example 1 and Example 2. The window is not time-based, but is based on the number of issues — in the charts above, it includes two issues ahead and two issues behind.

The rolling average starts at the first data point and the variations are easy to understand. In addition, it handles the outlier better. The rolling average is a more steady line towards the right of the chart, and clearly shows the last issue as an outlier with the appropriate standard deviation.

Cumulative Flow Diagram

Screenshot: Cumulative Flow Diagram

A Cumulative Flow Diagram (CFD) is an area chart that shows the various statuses of work items for an application, version, or sprint. The horizontal x-axis in a CFD indicates time, and the vertical y-axis indicates cards (issues). Each colored area of the chart equates to a workflow status (i.e. a column on your board).

The CFD can be useful for identifying bottlenecks. If your chart contains an area that is widening vertically over time, the column that equates to the widening area will generally be a bottleneck.

Viewing the Cumulative Flow Diagram

1. Navigate to your desired board.
2. Click **Reports**, then select **Cumulative Flow Diagram**.
   - To refine the data shown in the report, click **Refine report**, and select the desired filters.
   - To select a different timeframe, click the date range drop-down at the top of the chart.
   - To select a different date range, drag your cursor across the 'Overview' at the bottom of the chart.

*Tip: Click **How to read this chart** at the top of the report to view a short description of the report.*

Understanding the Cumulative Flow Diagram

Before you start using the CFD, you should get to know how it works. The following information will help you understand the key functionalities of the CFD:

- The CFD is board-specific – that is, it will only include issues that match your board's saved filter.
- The CFD is based on your board's column mapping. An issue is considered to be 'To Do' when it is in a status that has been mapped to the left-most column of your board. Similarly, an issue is considered to be 'Done' when it is in a status that has been mapped to the right-most column of your board. See Configuring columns for more information.

Next steps

> (i) **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

403

Epic Burndown

Screenshot: Epic Burndown report (story points)

About the Epic Burndown report

The Epic Burndown report shows you how your team is progressing against the work for a epic. An epic is a large user story that can be broken down into a number of smaller stories. The report will show data based on the estimation statistic that your board is using.

Here are some of the ways that you could use an Epic Burndown report:

- See how quickly your team is working through the epic,

On this page:

Jira Software Data Center and Server 9.12 documentation

2

- See how work added and removed during the sprint has affected your team's overall progress, and
- Predict how many sprints it will take to complete the work for an epic, based on past sprints and changes during the sprints.

- Understanding the Epic Burndown report
- Known issues
- Next steps

If you have used the Epic Report before, you will notice some similarities. However, the Epic Burndown is optimized for scrum teams that work in sprints — which makes tracking much easier.

Viewing the Epic Burndown report

1. Navigate to your desired board.
2. Click **Reports**, then select **Epic Burndown**.
3. Select the relevant epic from the Epic Burndown drop-down. You will be able to select from epics that are in projects configured for your board (via the board's filter).

*Tip: Click **How to read this chart** at the top of the report to view a short description of the report.*

⚠ *If you are using Internet Explorer 8, the Epic Burndown will not work.*

Printing the Epic Burndown report

To print the report, view the report and use the print functionality for your browser. The report will fit on either A4- or Letter-sized pages in both portrait and landscape modes (note, there is a known issue printing in landscape using Chrome).

Understanding the Epic Burndown report

Before you start using the Epic Burndown report, you should get to know how it works.

The sprint bar

- **Light green section** = work completed during the sprint.
  Note, if a bar is completely light green, you won't be able to tell how much of the work completed was originally estimated or not. To find out this information, click the bar to view the details.
- **Light blue section** = work that is remaining in the epic, out of the total work estimated for the epic at the start of the sprint.
- **Dark blue section** = work that was added during the sprint, but not originally included (i.e. scope change).
- **Light green section + light blue section** = total work in the epic that was originally estimated at the start of the sprint.
- **Light blue section + dark blue section** = total work in the epic that is remaining at the end of the sprint.
- **Bars with grey sections** = predicted sprints (see below).

Predicted sprints

Predicted sprints are calculated based on your team's velocity* (amount of work completed in the last three sprints), and the total work remaining for the epic. Scope change is not considered when calculating the velocity*, but is included in the total work remaining.

* *not the same as the velocity described in the Velocity Chart*

Consider the following example:

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

1. **Assessing the outstanding work:** *12 story points are remaining for the epic*, at the start of the current sprint (sprint 10).
2. **Calculating the velocity:** 7 story points were completed in the last three sprints (sprint 8, sprint 9, and sprint 10). This averages out to a velocity of *2 story points per sprint*, rounding to nearest story point.
3. **Predicting the remaining sprints:** At a velocity of 2 story points per sprint, it will take *6 more sprints* to complete the work for the epic. That is, 6 sprints of 2 story points each.

The current sprint is usually not counted when calculating the team's velocity. In the example above, the current sprint bar shows grey sections (like the bars for the predicted sprints) to represent this. The exception is when you have already *completed more work in the current sprint than the work that was predicted to be completed.* In this case, the current sprint (and the actual work completed) is used as one of the three sprints used to calculate the velocity. Also, the sprint bar will show green/blue sections, like the bars for completed sprints.

For example, in the chart above, if your team *had not* completed more than 2 story points in sprint 10, then the work completed in sprint 7, sprint 8, and sprint 9 would be used to calculate the velocity — rather than sprint 8, sprint 9, and sprint 10.

Other functionalities

The following questions and answers cover the other key functions of the Epic Burndown report:

- The **first sprint** shown is the one that contains the first issue (in the epic) that transitions out of the 'To Do' status, i.e. work is started on the epic.
- The **last sprint** shown is the one when all work is completed for the epic; or if work remains, then the predicted sprint when work will be finished.

The mapping of statuses to your board determines when an issue is considered 'To Do' or 'Done'. See Config uring columns for more information.
The Epic Burndown report can only make predictions based on the estimated issues in your epic. This does not include issues that cannot be estimated (e.g. you have configured an issue type to not have the Story Points field). If you have a high percentage of unestimated issues, then the predictions in the report will not be reliable (the **% unestimated issues** label is colored red when the percentage is above 30%).

For example, if you have only estimated 10% of the issues in the epic, then the report predicts the completion of work for the epic based on the 10% of the total issues. In reality, your team probably has much more work left to complete.
The following changes affect the original estimate of a sprint:

- An issue in an epic (before it started) is estimated (estimate is added)
- An issue in an epic (before it started) is re-estimated (estimate changes)

The following changes affect the scope of a sprint:

- An issue is added to an epic (after it was started) with an existing estimate
- An issue that was added to an epic (after was it started) is estimated (estimate is added)
- An issue that was added to an epic (after was it started) is re-estimated (estimate changes). Note, if the issue is re-estimated in a later sprint, the scope is retroactively adjusted in the sprint that the issue was originally added to.

Any change (burndown or scope) that occurs outside a sprint will be shown as part of the sprint with the latest start date before the change date.
Issue completed in a sprint, then reopened:

- The issue will not be shown in the earlier sprint.

Issue completed in an epic, but removed from the epic afterwards:

- The scope will remain unchanged and the work completed is still shown.

Issue completed in another epic, but later included in the epic (shown on the report):

- The scope will remain unchanged.

Issue completed in a sprint, but only added to the epic afterwards:

- The issue will be shown on the report, as if it was always part of the epic.

If your issue is in an unmapped status (i.e. status not mapped to a column), it will not be considered in the Epic Burndown report. That is, it won't be included in the sprint bars, the % unestimated issues, remaining story points, etc.

Known issues

If you encounter an issue that is not on this list, please raise it in our issue tracker.

key	summary	type	priority	status

⚠ Jira project doesn't exist or you don't have permission to view it.

View these issues in Jira

Next steps

ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Configuring working days
- Configuring estimation and tracking
- Epic Report
- Release Burndown

Epic Report

Epic Report ‑

ⓘ **How to read this chart**

Understand the progress towards completing an epic over time. This helps you manage your team's progress by tracking the remaining incomplete/unestimated work.

Hide this information

Bugfix and warranty issues ▾

TIS-100 Bugfix and warranty issues   Linked pages

Status Report

Completed Issues

Key	Summary	Issue Type	Priority
TIS-81	Afterburner initial design	🔲 Bug	↑ Critica
TIS-82	Afterburner initial prototype	🔲 Bug	↓ Minor

Screenshot: Epic Report

The Epic Report shows a list of complete, incomplete, and unestimated issues in an epic. It is particularly useful in planning work for an epic that may extend over multiple sprints.

Use the Epic Report to understand the progress towards completing an epic over time, and to track the amount of remaining work that's incomplete or unestimated.

Before you begin

- This page only applies to Scrum boards.

On this page:

Viewing the Epic Report

1. Navigate to your desired board.
2. Click **Reports**, then select **Epic Report**.
3. Select the relevant epic from the epic drop-down.
   - Click **View in Issue Navigator** to jump to the Issue Navigator and see a list of all the epic's issues.

*Tip: Click **How to read this chart** at the top of the report to view a short description of the report.*

Understanding the Epic Report

Before you start using the Epic Report, you should get to know how it works. The following information will help you understand the key functionalities of the Epic Report:

- The Epic Report is based on your board's column mapping. An issue is considered to be 'To Do' when it is in a status that has been mapped to the left-most column of your board. Similarly, an issue is considered to be 'Done' when it is in a status that has been mapped to the right-most column of your board. See Configuring columns for more information.
- The graph will look different if you are using Issue Count as your Estimation Statistic (rather than Story Points, as shown in the screenshot above).

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Working with epics
- Configuring estimation and tracking

Release Burndown

Screenshot: Release Burndown report (story points)

About the Release Burndown report

The Release Burndown report shows you how your team is progressing against the work for a release. In Jira Software, there is no 'release' entity — a version is equivalent to a release (hence, the term 'version' will be used instead of 'release' in this document). The report will show data based on the estimation statistic that your board is using.

Here are some of the ways that you could use a Release Burndown report:

- See how quickly your team is working through the backlog,
- See how work added and removed during the sprint has affected your team's overall progress, and
- Predict how many sprints it will take to complete the work for a version, based on past sprints and changes during the sprints.

If you have used the Version Report before, you will notice some similarities. However, the Release Burndown report is optimized for scrum teams that work in sprints — which makes tracking much easier.

Viewing the Release Burndown report

1. Navigate to your desired board.
2. Click **Reports**, then select **Release Burndown**.
3. Select the relevant version from the Release Burndown drop-down. You will be able to choose from versions that are in projects configured for your board (via the board's filter).

*Tip: Click **How to read this chart** at the top of the report to view a short description of the report.*

⚠️ *If you are using Internet Explorer 8, the Release Burndown will not work.*

Printing the Release Burndown report

To print the report, view the report and use the print functionality for your browser. The report will fit on either A4- or Letter-sized pages in both portrait and landscape modes (note, there is a known issue printing in landscape using Chrome).

Understanding the Release Burndown report

Before you start using the Release Burndown report, you should get to know how it works.

The sprint bar

- **Light green section** = work completed during the sprint.
  Note, if a bar is completely light green, you won't be able to tell how much of the work completed was originally estimated or not. To find out this information, click the bar to view the details.
- **Light blue section** = work that is remaining in the version, out of the total work estimated for the version at the start of the sprint.
- **Dark blue section** = work that was added during the sprint, but not originally included (i.e. scope change).
- **Light green section + light blue section** = total work in the version, that was originally estimated at the start of the sprint.
- **Light blue section + dark blue section** = total work in the version, that is remaining at the end of the sprint.
- **Bars with grey sections** = predicted sprints (see below).

Predicted sprints

Predicted sprints are calculated based on your team's velocity* (amount of work completed in the last three sprints), and the total work remaining in your backlog. Scope change is not considered when calculating the velocity*, but is included in the total work remaining.

* not the same as the velocity described in the Velocity Chart.

Consider the following example:

411

1. **Assessing the outstanding work:** *26 story points are remaining for the version*, at the start of the current sprint (sprint 5).
Note, the '22 remaining (story points)' label (at the top of every Release Burndown report) subtracts the 4 story points that are predicted to be completed in the current sprint.
2. **Calculating the velocity:** 11 story points were completed in the last three sprints (sprint 2, sprint 3, and sprint 4). This averages out to a velocity of *4 story points per sprint*, rounding to nearest story point.
3. **Predicting the remaining sprints:** At a velocity of 4 story points per sprint, it will take *7 more sprints (including the current sprint)* to complete the work for the version: 26 story points. That is, 6 sprints of 4 story points, plus one final sprint of 2 story points.

The current sprint is usually not counted when calculating the team's velocity. In the example above, the current sprint bar shows grey sections (like the bars for the predicted sprints) to represent this. The exception is when you have already *completed more work in the current sprint than the work that was predicted to be completed*. In this case, the current sprint (and the actual work completed) is used as one of the three sprints used to calculate the velocity. Also, the sprint bar will show green/blue sections, like the bars for completed sprints.

For example, in the chart above, if your team had already completed more than 4 story points in sprint 5, then the work completed in sprint 3, sprint 4, and sprint 5 would be used to calculate the velocity, rather than sprint 2, sprint 3 and sprint 4.

Other functionalities

The following questions and answers cover the other key functions of the Release Burndown report:

The **Start date** and **Release date** configured for the version are shown on the bottom of the report as the **Pla nned start date** and **Planned end date**. However, these are *planned* dates, and they do not determine the first and last sprints shown on the report.

- The **first sprint** shown is the one that contains the first issue (in the version) that transitions out of the 'To Do' status, i.e. work is started on the version.
- The **last sprint** shown is the one when all work is completed for the version; or if work remains, then the predicted sprint when work will be finished.

The mapping of statuses to your board determines when an issue is considered 'To Do' or 'Done'. See Config uring columns for more information.
The Release Burndown report can only make predictions based on the estimated issues in your version. If you have a high percentage of unestimated issues, then the predictions in the report will not be reliable (the **% unestimated issues** label is colored red when the percentage is above 30%, to help make you aware of this).

For example, if you have only estimated 10% of the issues in the version, then the report predicts the completion of work for the version based on the 10% of the total issues. In reality, your team probably has much more work left to complete.
The following changes affect the original estimate of a sprint:

- An issue in a version (before it started) is estimated (estimate is added)
- An issue in a version (before it started) is re-estimated (estimate changes)

The following changes affect the scope of a sprint:

- An issue is added to a version (after it was started) with an existing estimate
- An issue that was added to a version (after was it started) is estimated (estimate is added)
- An issue that was added to a version (after was it started) is re-estimated (estimate changes). Note, if the issue is re-estimated in a later sprint, the scope is retroactively adjusted in the sprint that the issue was originally added to.

Any change (burndown or scope) that occurs outside a sprint will be shown as part of the sprint with the latest start date before the change date.
Issue completed in a sprint, then reopened:

- The issue will not be shown in the earlier sprint.

Issue completed in a version, but removed from the version afterwards:

- The scope will remain unchanged and the work completed is still shown.

Issue completed in another version or without a version, but later included in the version (shown on the report):

- The scope will remain unchanged.

Issue completed in a sprint, but only added to the version afterwards:

- The issue will be shown on the report, as if it was always part of the version.

If your issue is in an unmapped status (i.e. status not mapped to a column), it will not be considered in the Release Burndown chart. That is, it won't be included in the sprint bars, the % unestimated issues, remaining story points, etc.

Known issues

If you encounter an issue that is not on this list, please raise it in our issue tracker.

key	summary	type	priority	status

⚠ Jira project doesn't exist or you don't have permission to view it.

View these issues in Jira

Next steps

ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Planning a version
- Configuring working days
- Configuring estimation and tracking
- Version Report
- Epic Burndown

Sprint Report

The Sprint Report shows the list of issues in each sprint. It is useful for your Sprint Retrospective meetings, and also for mid-sprint progress checks.

Screenshot: Sprint Report

Tip: If you have Jira Software connected to Confluence, you can create a 'Retrospectives' page via the **Linked pages** *link. See* Linking a Confluence page to a sprint *for details.*

Before you begin

- This page only applies to Scrum boards.

Viewing the Sprint Report

1. Navigate to your desired board.
2. Click **Reports**, then select **Sprint Report** .
3. Select the relevant sprint from the sprint drop-down.

Tip: Click **How to read this chart** *at the top of the report to view a short description of the report.*

Understanding the Sprint Report

Before you start using the Sprint Report, you should get to know how it works. The following information will help you understand the key functionalities of the Sprint Report:

- The Sprint Report is board-specific – that is, it will only include issues that match your board's saved filter.
- The Sprint Report includes only issues that were completed while the sprint was in progress.
- Issues added after the sprint starts are indicated with an asterisk.
- An issue is considered to be 'To Do' when it is in a status that has been mapped to the left-most column of your board. Similarly, an issue is considered to be 'Done' when it is in a status that has been mapped to the right-most column of your board. See Configuring columns for more information.
- The grey line is a guideline that's drawn from the total estimate of the issues at the start of the sprint, to 0 at the end of the sprint. The grey line stays flat during non-working days.
- The red line represents actual work done during the sprint — it's drawn from the total estimate of the issues at the start of the sprint, and it goes higher or lower as your team works on the issues in the sprint. Note that the red line shows the current total estimate for unresolved issues at any point in the sprint. It also **reflects issues that are added or removed from the sprint.**
- If your board is configured to track remaining estimates and time spent, a green line will display in the Sprint Report, indicating work logged on issues.
- You can monitor the actual vs the planned start and end dates for your sprint.

What can I do in the Sprint Report?

You can complete the following actions for any selected sprint. Make sure you choose the right sprint from the drop-down list.

Task	Instructions
Reopen a sprint	Click ⋯ **> Reopen sprint**. The sprint, and its issues, will be moved back to the scrum board.
Edit a sprint	Click ⋯ **> Edit sprint** to edit a sprint's name or goal.
Delete a sprint	Click ⋯ **> Delete sprint**. The sprint, and its issues, will be deleted forever.
View a sprint in the issue navigator	Click ⋯ **> View sprint in issue navigator**.

Next steps

> ⓘ **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Planning sprints
- Completing a sprint
- Configuring working days

Velocity Chart

A team's velocity is the amount of value it can deliver in each sprint. Whether your team is using story points, issue count, or hours to estimate their stories, the chart will show how well you're keeping up with your commitments. Over time, you'll be able to predict the amount of work you can realistically commit to and get done in future sprints, making sure you're not running around in panic.

Viewing the Velocity Chart

The chart is available for all Scrum boards and requires that you've completed at least one sprint to show any meaningful data.

1. Navigate to your desired board.
2. Click **Reports**, then select **Velocity Chart**.
3. The chart will be displayed. By default, it shows all sprints completed in the past 3 months, but you can change this timeframe.

1. **Timeframe:** Choose one of the predefined timeframes, or select a date range. Whatever you choose, it will be retained in the URL, so you can share the link with your teammates.
2. **Sprint details:** Hover over a sprint to see the details, and the work committed and completed. This gives you an idea of whether your team is over- or under-committing. Ideally, the two bars for each sprint should be the same.
3. **Save as image:** Click to save the chart for future reference.
4. **Focus:** Use focus to zoom in on the sprints you're most interested in.

Good to know

Here are some things to keep in mind when viewing the chart.

	Details
Boards	The chart is board-specific, and will only include estimates from issues that match your board's saved filter.
Sprints	The chart includes only completed sprints whose *End date* is within your chosen timeframe. By default, it shows the maximum of 120 sprints—if your timeframe includes more, we'll show the latest 120.

Issues	The chart includes estimates (committed and completed) from all issues in your sprints. For performance reasons, we're showing the maximum of 25,000 issues.
Permissions	When a user is viewing the chart, they will only see the estimates from the issues they're allowed to see. It might happen that the chart is showing different data for different users, because they don't have the right permissions.
Estimation	The estimates shown on the chart can be based on story points, hours, issue count, business value, or any numeric field of your choice. For more information, see Configuring estimation and tracking.

Using viewing options to hide unrelated sprints

The chart has a *Viewing options* section that allows you to hide unrelated sprints. This is useful to make sure that the chart includes only sprints that really matter for your team.

For example, if an issue in *Sprint 1* on your board was moved to *Sprint 2* on a different board, *Sprint 2* would be included in the chart. This might affect your team's velocity, because *Sprint 2* belongs to a completely different board (and most likely a different team, too). With this option enabled, only sprints that belong to your desired board are included in the chart.

Next steps

> (i) **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Configuring estimation and tracking

Version Report

Reports: Version Report ▾

ⓘ **How to read this chart**

Track the projected release date for a version. This helps you monitor whether the version will release on time, so you can take action if work is falling behind.

Hide this information

Version Report   TIS: 3.0 ▾

Scheduled Release Date: 14/Sep/15

Status Report

Completed Issues

Key	Summary	Issue Type	Priority	S
TIS-23	Engage JetShuttle SpaceWays for short distance space travel	⊞ Travel Provider	↑ Major	
TIS-45	Email non registered users to sign up with Teams In Space	🟡 Story	↑ Critical	
TIS-20	Engage Saturn Shuttle Lines for group tours	⊞ Travel Provider	↑ Major	

Screenshot: Version Report

The Version Report shows your team's progress towards the completion of a version. The Version Report also shows you the predicted Release Date, based on your team's average rate of progress (velocity) since the start of the version, and the estimated amount of work remaining.

Before you begin

- This page only applies to Scrum boards.

Viewing the Version Report

1. Navigate to your desired board.
2. Click **Reports**, then select **Version Report**.
3. Select the relevant version from the Version Report drop-down.

*Tip: Click **How to read this chart** at the top of the report to view a short description of the report.*

Understanding the Version Report

Before you start using the Version Report, you should get to know how it works. The following information will help you understand the key functionalities of the Version Report:

- The Version Report is board-specific – that is, it will only include issues that match your board's saved filter.
- The Version Report will exclude issues of a 'sub-task type'.
- The Version Report shows 'Released' versions but not 'Archived' versions. For more about version status, see Managing versions (*Jira Admin documentation*).
- The Version Report is based on your board's column mapping. An issue is considered to be 'To Do' when it is in a status that has been mapped to the left-most column of your board. Similarly, an issue is considered to be 'Done' when it is in a status that has been mapped to the right-most column of your board. See Configuring columns for more information.
- The horizontal axis starts on the version's Start Date; or if no Start Date is specified, the date on which an issue was first added to the version. The graph shows the state your version was in at any given point in time, in terms of your total and completed Story Points (or other Estimation Statistic of your choice), so that you can see how the scope may have changed, and how you are progressing towards completion of the estimated work.
- The graph shows you the following predictions:
  - the **Predicted Release Date** *(blue line)* – that is, the date at which you can expect all issues in your version to be complete, based on your average daily velocity and the amount of estimated work remaining.
  - the **Predicted Release Date (Optimistic)** *(shaded area to the left of the blue line)* – that is, the earliest date by which you might expect the version to be complete. (The 'optimistic' date is calculated by adding 10% to the average daily velocity.)
  - the **Predicted Release Date (Pessimistic)** *(shaded area to the right of the blue line)* – that is, the latest date by which you might expect the version to be complete. (The 'pessimistic' date is calculated by subtracting 10% from the average daily velocity.)
- 10% of the estimated work for the version will need to be complete before the predictions can be calculated.

Next steps

> (i) **Need help?** If you can't find the answer you're looking for in our documentation, we have other resources available to help you. Check out Getting help.

Read the following related topics:

- Planning a version
- Configuring working days
- Configuring estimation and tracking

Configuring dashboards

Your dashboard is the main display you see when you log in to your project. You can create multiple dashboards for different projects, or multiple dashboards for one big project. Each project has a default dashboard, or you can create a personal dashboard and add gadgets to keep track of assignments and issues you're working on. Dashboards are designed to display gadgets that help you organize your projects, assignments, and achievements in different charts.

You can see all dashboards by selecting the **Dashboards** drop-down from your Jira application header.

On this page:

- About the default dashboard
- Creating a dashboard
- Managing dashboards and permissions
- Sharing and editing your dashboard
- Adding favorite dashboards
- Note on dashboard permissions
- Setting up a Wallboard

About the default dashboard

The gadgets on the default dashboard can be reordered and switched between the left and right columns. Additional gadgets can also be added, while some gadgets can be configured. The layout of the dashboard (e.g. number of columns) can also be configured.

All changes made to the default dashboard will also change the dashboards of all users currently using the default dashboard. However, gadgets that users do not have permissions to see will not be displayed to them. For example, the 'Administration' gadget, although it may exist in the default dashboard configuration, will not be visible to non-admin users.

Creating a dashboard

You can easily create and customize your own dashboard to display the information you need. Note that only administrators can customize the default dashboard for your project.

1. At the top right of the Dashboard, select **More** (...).
2. Select either **Create Dashboard** to create a blank dashboard, or **Copy Dashboard** to create a copy of the dashboard you are currently viewing.
3. Name and describe your dashboard.
4. Fill out the rest of the fields as applicable.
5. Click **Add**.

By default, sharing is set to private if you have not specified a personal preference. You can adjust this setting in the sharing preferences in your user profile, and change dashboard permissions at any time in the Manage Dashboards page.

Choosing a dashboard layout

To choose a different layout for your dashboard page (for example, three columns instead of two):

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** (⬚) > **Vie w 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 Gad gets 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, you'll find all the gadgets that you can install on dashboards in Jira Core (JC), Jira Software (JSW), and Jira Service Management (JSM), including additional Bamboo, Crucible, and FishEye gadgets.

Gadget	JC	JSW	JSM	Description
Jira Core gadgets				
Assigned to Me	✅	✅	✅	Shows all unresolved issues assigned to you.
Bubble Chart	✅	✅	✅	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 field's 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.
Resoluti on 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 isn't set, the gadget won't 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).
Wallboar d 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 can't modify the gadget's color as it matches the black background color in the wallboard view.
Jira Software gadgets				
Agile Wallboar d Gadget		✅		Helps you see how you're tracking with an agile board on your wallboard or dashboard.
Days Remaini ng in Sprint Gadget		✅		Shows how many working days you have before the current sprint ends.
Sprint Burndow n 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 doesn't 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 won't be altered. If you're using Jira's time tracking, the scope change won't 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.

Rich Text	✅	✅	✅	The Rich Text gadget has substituted the deprecated Text gadget. The Rich Text gadget uses the rich text editor instead of the plain textarea in edit mode. The gadget doesn't contain any arbitrary HTML and won't expose the system to any security risks.
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 config ure 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 Summar y Chart	✅	✅	✅	Shows a graphical summary of a Bamboo build plan. If you want to add the gadget to the dashboard, a Jira admin should config ure 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 config ure 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 config ure 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. |

Using Jira on a mobile device

> ⚠ We've deprecated the Jira mobile web interface with the release of Jira 9.11. We'll remove it in Jira 9.12 LTS. For a richer experience, try the Jira Data Center and Server mobile app for 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 the Jira desktop interface via the mobile menu (shown in the screenshots below).

What does Jira look like on a mobile device?

What can you do in Jira on a mobile device?

The Jira mobile interface has been designed to give users quick access to their issues on the go. This includes;

- Viewing issues, comments, attachments, issue links and your favorite filters.
- Performing basic operations like adding comments, watching or voting on issues and assigning issues to users.

If you need to create or modify issues on the go, you can still do so by switching to the desktop interface via the mobile menu (shown in the screenshots above).

Discover Advanced Roadmaps for Jira

> ⓘ Advanced Roadmaps is now part of Jira Software Data Center. To get access to the features described on this page, upgrade to Jira Software Data Center 8.15 or later. Can't upgrade yet? Depending on your current Data Center version, you can access these features by installing the latest version of the app (at no cost). See our FAQ for all the details

Advanced Roadmaps is a powerful roadmapping solution for big picture planning. Available as part of Jira Software Data Center, you can plan and track work across multiple projects and teams. Advanced Roadmaps accesses boards, projects, and filters in Jira Software to visualize data in a customizable interface. Functioning as a sandbox environment, you can plan and experiment before updating your original data in Jira Software.

As of February 2021, access to Advanced Roadmaps is included as part of your Jira Software Data Center license. To get started with Advanced Roadmaps, open Jira Software and go to the **Plans** tab.

Advanced Roadmaps works best in Chrome or Firefox.

If you haven't upgraded recently, you might be eligible to get Advanced Roadmaps for free. For more information , see our FAQ page.

> Advanced Roadmaps is compatible with the use of all currently supported versions of Jira Core and Jira Software, as noted in the Atlassian Support End of Life Policy.

Get started with Advanced Roadmaps

> (i) Advanced Roadmaps is now part of Jira Software Data Center. To get access to the features described on this page, upgrade to Jira Software Data Center 8.15 or later. Can't upgrade yet? Depending on your current Data Center version, you can access these features by installing the latest version of the app (at no cost). See our FAQ for all the details

Advanced Roadmaps is a powerful planning tool that helps you visualize the work needed to meet your goals. However, before you can wow the world with your inspired plans, you need to set up your Advanced Roadmaps instance. In this section, we'll cover the basics of how to set up your planning environment, and what settings you can change to better suit how your organization gets work done.

Set up Jira Software

You can use Advanced Roadmaps with minimal setup. However, some tasks, such as using custom fields, surfacing the Teams field in your Jira issues, and tracking issues by releases, need to be configured by an administrator.

Detailed instructions can be found on the Configure your Jira Software instance page.

Configure your plan

Advanced Roadmaps has a range of options and settings so you can create a plan that matches the ways your teams work. You can change how issues are estimated, which dates Advanced Roadmaps uses as default when scheduling, configure dependencies and hierarchies, and more. With the exception of estimation units and date settings, these changes will be applied to all plans in your instance as well as those that are yet to be created.

Learn how to change these settings and more on the Configure your Advanced Roadmaps plan settings page.

Add custom fields

Advanced Roadmaps lets you import most custom fields you may be using in Jira Software to use in your plan. The following custom field types are supported:

- checkbox
- date picker
- user picker (single user)
- number
- radio button
- label
- single-choice select
- multiple-choice select
- single-line text
- URL (read-only)

To learn more about how to configure these, see our page on custom fields.

Configure permissions in Advanced Roadmaps

Lastly, administrators can configure global permissions and restrict access to plans based on users or groups. Advanced Roadmaps adheres to both sets of permissions, but plan permissions take precedence over global.

Learn how to configure Permissions in Advanced Roadmaps.

How Advanced Roadmaps differs in Data Center and Server

Jira Data Center and Jira Server both offer on-prem solutions for organizations that want to host their own Jira instance. Learn more about the differences between Server and DC editions of Jira Software.

Advanced Roadmaps is included as part of Jira Data Center from version 8.15 onwards. This version sees some feature updates, but we recommend migrating to Jira Software Cloud to get the most up-to-date planning features. Learn more about the features available on Jira Software Cloud.

Configure your Jira Software instance for Advanced Roadmaps

> ⓘ This page is for those with Jira Software administrator access only.

Advanced Roadmaps is fully functional and compatible with Jira Software out-of-the box, but certain custom fields and features need to be configured by your administrator. If you decide not to configure these fields, you can still use them in Advanced Roadmaps, but you won't be able to save the changes made in these fields back to your Jira issues.

Jump to:

- Surface fix versions in Jira issues
- Enable custom fields used in Advanced Roadmaps in Jira
- Configure custom hierarchy levels
- Configure Dependencies in Jira for Advanced Roadmaps

Surface fix versions in Jira Software issues

What are known as releases in Advanced Roadmaps are known and treated as fix versions in Jira Software. If the fix version field is hidden in your issues, then the releases you set in your plan won't display in the corresponding issues.

To display the fix version field in Jira Software:

1. Navigate to the Jira Software admin panel using the ⚙ icon in the upper right and choose **Issues**.
2. From the menu on the left, select **Field Configurations**, then the field configuration of interest.
3. Select **Configure**, then find the Fix Version field.
4. In the Actions column, select **Show**.

Each time a release is set for an issue in your plan, and you save this change to Jira Software, the release will appear as the fix version for the corresponding issue

Enable custom fields used in Advanced Roadmaps in Jira

Before you can use labels, components, team, and custom dates in Advanced Roadmaps, they first must be defined in your Jira settings.

Only the following custom field types are supported in Advanced Roadmaps:

- label
- component
- teams
- checkbox
- date picker
- user picker (single user)
- number
- radio button
- single-choice select
- multiple-choice select
- single-line text
- URL (read-only)

Adding these fields is a two step process:

1. Add custom fields to your Jira issues

1. Select ⚙ > **Issues**.
2. Under **FIELDS**, select **Custom fields**.
3. Click **Create custom field**. Make sure to select **All** to view the available field options.
4. Select the type of field you want to create and click **Next**.
5. Add the details for your field.
    a. The name you select appears as the custom field's title wherever the field appears. The description appears below the field when creating and editing issues, but not when browsing issues.
6. When you have entered the field details, select **Create**.
7. Add the new custom field to one or multiple screens by ticking the corresponding box and click **Update**.

Data in these custom fields will be saved to your Jira issues, but won't be visible until the fields are added to your instance.

Optional: Once you've created your custom field, you can configure it have have a default value. Learn how to c onfigure default values in custom fields in Jira.

2. Add custom field to issue scheme

1. Select ⚙ > **Issues.**
2. In the left menu, select **Custom fields** listed under **Fields**.
3. Find your recently added field, then select > **Associate to Screens**.
4. Select the appropriate screen(s), then scroll to the bottom and select **Update**.

Now that you've added the custom field to issues and the scheme, Advanced Roadmaps users can now add it to their plans. Learn how to add it to on the Add and view custom fields in Advanced Roadmaps page.

Configure custom hierarchy levels

Before you can use custom hierarchy levels in Advanced Roadmaps, the issue type must first be created in Jira by an administrator. Once that's done, the issue type must be associated to your Advanced Roadmaps hierarchy.

This is a three step process:

1. Create issue type in Jira Software (if it doesn't already exist)

1. Select ⚙ > **Issues**.
2. In the left column, go to **Issue types** > **Add issue type**.
3. Enter a name and description for your new issue type.
4. Choose between a standard or sub-task issue type.
5. Standard issue types are placed above the epic level (commonly Initiative and Legend) whereas Sub-task issue types are underneath the Story level alongside subtasks.
6. Select **Add**.

2. Add issue type to scheme in Jira Software

1. Select ⚙ > **Issues.**
2. From the left column, select **Issue type schemes**.
3. Find the relevant issue type scheme and click **Edit**.
4. Edit the name, description, default issue type, and selected issue types.
5. To add an issue type to the scheme, drag it from **Available Issue Types** to **Issue Types for Current Scheme**.
6. When you're done, select **Save**.

3. Add custom hierarchy levels to your Advanced Roadmaps plan:

1. Select ⚙ > **Manage apps**.
2. In the menu on the left, select **Advanced Roadmaps hierarchy configuration.**
3. Select or remove hierarchy levels you want to include in your plans.
4. Select **Save changes**.

Your custom hierarchy levels should now be available for you to use in Advanced Roadmaps.

Configure Dependencies in Jira for Advanced Roadmaps

By default, dependencies in Advanced Roadmaps use the **Blocks** issue link type. In order to use custom issue links in your Advanced Roadmaps plan, they must be created and defined in Jira, then added to Advanced Roadmaps.

This is a two step process:

1. Configure your issue link in Jira Software

1. Navigate to the Jira Software admin panel using the ⚙ icon in the upper right and select **Issues**.
2. In the left hand sidebar, select **Issue linking** under **Issue features**.
3. Make sure that Issue linking is set to **ON**. If it's not, select **Activate** to enable it.
4. Once activated, you can edit the **Name**, **Outward description**, and **Inward description** of an existing issue link by selecting **Edit** from the right column. Use **Update** to save your changes.
5. To add a new link, define the name of your new issue link and the descriptions using the fields under the **Add New Link Type** header. Use **Add** to save it.

2. Apply that configuration to Advanced Roadmaps

1. Navigate to the Jira Software admin panel using the ⚙ icon in the upper right and select **Manage apps**.
2. Go to Advanced Roadmaps **dependencies**. Add an issue linking type using the dropdown menu, or remove it with the **x** that appears in the box. Use **Swap** on the right side to invert the dependent relationship.
3. Select **Save changes**.

These changes will apply to dependencies in all plans in your instance.

You also have the option of deciding whether Advanced Roadmaps uses concurrent or sequential dependencies. To read more about what these mean, see the Use sequential or concurrent dependencies in your plan page.

To change between sequential and concurrent dependencies:

1. From the **Roadmap** view, navigate to settings using the next to your plan name > **Configure**.
2. Under **Dependencies**, select either sequential and concurrent dependencies.

Once configured, learn how to use dependencies in your plan.

Configure your Advanced Roadmaps plan settings

> ℹ️ For all of the following procedures, you must have the **Advanced Roadmaps admin** access.

As the administrator, you get to define how Advanced Roadmaps handles almost every aspect of your plan, from where it pulls issues to how it treats dependencies.

In this section, we'll discuss how to adjust the following:

- Issue sources
- Estimation units
- Dates
- Dependencies
- Auto-scheduler

Issue sources

To create and maintain an accurate plan, you must first specify in the Advanced Roadmaps where to find your issues in Jira Software. This is referred to as an issue source and is first defined while creating a new plan, though you can change or add more issue sources after plan creation.

To add or change the issue sources of your plan:

1. From the **Plans** tab, select **View plans,** and open the plan you want to configure.

2. Go to **Settings** ⚙ > **Configure.**

3. Select **Issue sources**. You can select a [project](#), [board](#), or [filter](#) as the source of issues for your plan.

Estimation units

You can choose which unit to use when estimating issues in your plan. Estimation for issues can be based on time (hours or days) or [story points](#).

To change the estimation units in your plan:

1. From the Roadmap tab of your plan, navigate to **Settings** ⚙ > **Configure**.
2. In the **Scheduling** tab, select whether you'd like your plan to be estimated in **Hours**, **Days**, or **Story Points**.

> ⊖ Story points and time-based estimates (hours and days) are not interchangeable. For this reason, changing this setting will erase any issue estimations not already saved in Jira Software.

Time-based estimates

Time-based estimates can be used for both Scrum or Kanban teams. The default capacity is 200 weekly hours, though this can be changed. See our section on [Capacity](#) to learn how.

For Scrum teams, the default iteration length is set to two weeks. For Kanban teams, each iteration lasts one week.

Story points

If you're using story points to estimate issues, you can only use Scrum teams in your plan. The default team velocity is set to 30 story points, and the default iteration length is set to 2 weeks.

Dates

By default, target start and target end dates are used when scheduling issues in a plan. To align with how your teams work, you can use due dates, sprint dates, or custom dates to schedule issues.

Target start and target end dates are independent from local timezone settings. Due and custom dates are based on the local timezone settings of the user. However, if you're using due dates and custom dates as the start and end dates in the plan, users in different timezones will see the same dates in Advanced Roadmaps and different dates in Jira Software. This is becauseJira Software and Advanced Roadmaps handle dates and timezone settings differently.

To change the date settings in your plan:

1. From the Roadmap tab, navigate to **Settings** ⚙ > **Configure**.
2. In the **Scheduling** tab, the section titled **Dates** lets you select howAdvanced Roadmaps handles dates in your plan.

Custom dates

To use custom dates in your plan, you must add the [custom date field to the schemes of all the projects](#) associat ed with the plan's issue sources. Once added, these custom dates will be available in the dates settings (as outlined above) along with their corresponding IDs for easier identification.

If you want to use the Date Picker custom field, you make the following changes so that it's available in your plan:

- associate the field with all projects in a plan
- associate the field with all issue types in the Advanced Roadmaps for Jira (ARJ) hierarchy
- ensure that it has the Date Picker type

On your timeline, the custom dates will be displayed with the date lozenge (D) in the date column.

Sprint dates

> ⓘ This only applies to Scrum teams.

If an issue is assigned to a sprint, it can be scheduled according to the sprint's start and end dates. These dates will be displayed in the date fields section with the sprint lozenge (S). The issue will also have a schedule bar in the timeline, based on the sprint start and end dates.

To use sprint dates for issues without dates:

1. From the **Roadmap** tab, navigate to **Settings** ⚙ > **Scheduling**.
2. Under **Dates**, select the checkbox labeled **Use sprint dates when issues don't have start and end dates**.

Dependencies

When one issue must be completed before the other, or the progress of one issue is reliant on another issue's progress, this is referred to as a dependency. There are two types used in Advanced Roadmaps:

- Incoming dependencies are waiting on another issue to be completed before work can begin.
- Outgoing dependencies need to be completed before the next issue can begin.

These can be further configured by your admin. Learn how to configure dependencies on the Configure your Jira Software instance for Advanced Roadmaps page.

> ⓘ Dependencies settings will be applied to all plans in your Jira instance.

Sequential and concurrent dependencies

You can choose whether you want to work on dependent issues in order or at the same time. This is what Advanced Roadmaps refers to as sequential or concurrent dependencies. Sequential dependencies can't be worked on at the same time, while concurrent ones can. This is true whether the issues are manually scheduled or auto-scheduled.

To change between sequential and concurrent dependencies:

1. From the **Roadmap** tab, navigate to **Settings** ⚙ > **Scheduling**.
2. Under **Dependencies**, select either sequential and concurrent dependencies.

Auto-scheduler

Those with administrator permissions can configure which fields the auto-scheduler has permission to overwrite. They can also decide whether to only auto-fill empty fields or to fill all fields in a plan. These settings are available to administrators when they select the **Auto-schedule** tab from the plan view.

Unestimated issues that have sprint, release, or team values will not be overwritten regardless of your auto-schedule settings.

Configure custom fields in Advanced Roadmaps

Not all of the custom fields that are available in Jira Software can be used in Advanced Roadmaps. The following custom field types are supported:

- checkbox
- date picker
- user picker (single user)
- number
- radio button
- label
- single-choice select
- multiple-choice select
- single-line text
- URL (read-only)

When using label custom field types, you can only select existing labels. If you need to create a new label, you must do it in Jira Software.

Add and view custom fields

To add and view custom fields that are currently in your plan:

1. From the Roadmap tab, navigate to **Settings** ⚙ > **Configure** > **Custom fields**. This will show you all of the custom fields currently in your plan.
2. Select **Add custom field** in the upper right corner, then search for the fields you want to add.
3. Use the **Add fields** button to finalize your choice.

Delete custom fields

To remove a custom field, navigate to the custom fields menu using the steps outlined above, then use the **trash can icon** in the right-hand column. Any unsaved changes that correspond to this field will also be removed from your plan. You won't be able to delete any date picker custom fields that are used to schedule issues in the plan.

Permissions in Advanced Roadmaps

Advanced Roadmaps uses two different sets of permissions when allowing access to plans. The first is Global permissions which are defined by and inherited from Jira Software. By default, those with Jira Software administrator and system administrator access will have administrator access in Advanced Roadmaps.

You can also grant access to individual users on a per-plan basis within Advanced Roadmaps. These permissions only apply to the individual plan, and will overrule any global settings.

Grant access to your plan

To manage your plan permissions within Advanced Roadmaps:

1. In the Roadmap tab, select **Settings** ⚙ > **Configure**.
2. On the left sidebar, navigate to **Permissions**. You can grant view and edit permissions to individual users or groups, or make it visible to all Advanced Roadmaps users.

> ⓘ Private plans are visible to users with any of these permissions:
>
> - Advanced Roadmaps administrator
> - Jira Software administrator
> - System administrator

Permission types in Advanced Roadmaps

There are five different levels of access for Advanced Roadmaps users:

Advanced Roadmaps administrator

Users with this permission can perform all administrative functions in Advanced Roadmaps, including accessing any private plans. By default, Jira Software administrators are automatically made administrators in Advanced Roadmaps.

Advanced Roadmaps user

Users with this permission can create plans and view plans they have access to. For plans they can access, they can also edit their issue sources, edit any issues in the plans, and save these edited issue details in Jira Software.

Advanced Roadmaps viewer

Users with this permission can browse and view plans in read-only mode. In Jira Software, they can also update parent links, teams, and target dates of issues.

Advanced Roadmaps Labs access

Users with this permission can access and use the features of Advanced Roadmaps that are in Labs mode. When features in Labs mode are enabled, users have early access to the new features that are still being worked on.

Advanced Roadmaps shared team management

Users with this permission can access global team management and edit teams that are shared across plans.

Change global permissions

Global permission settings allow administrators to define how organization-wide access levels apply to Advanced Roadmaps. For example, a global permission might allow those with Jira administrator access to be automatically granted Advanced Roadmaps administrator access, too.

To configure global permissions for Advanced Roadmaps:

1. Navigate to the Jira Software admin panel using the ⚙ icon in the upper right and select **Manage apps**.
2. Find Permissions underneath the header **Access control**.
3. Change the user permissions as you want.
4. Select **Save changes**.

> ⓘ You must have the Jira Software administrator permission to change global settings.

Early access features in Advanced Roadmaps

ⓘ Enabling the features listed on these pages requires the **Jira Administrator** permission

To enable or disable early access features in your instance:

1. Select **Plans** in the ⚙ menu, then select **Administration**.
2. In the left sidebar, click **Plans early access features**.
3. Use the checkbox next the feature that you want to enable or disable in your instance.

Early access features

The following extra features can be enabled in this version of Advanced Roadmaps for Jira:

Plan default to system timezone

Enable this feature to set the system timezone as the default setting across all plans in Advanced Roadmaps.

Plan your work in Advanced Roadmaps

Simply put, a plan in Advanced Roadmaps contains all of the work that needs to be done on your project in a timeline format (also called a Gantt chart). Every piece of work is broken down into issues which can then be assigned to teams.

A plan or roadmap contains the following items to help you map your upcoming work:

- **Issues** - These are the core building blocks of your plan. each issue represents a task in Jira that must be completed. These issues can be assigned a work estimate, given start and end dates, nested under parent issues, and assigned to individuals or teams. Learn how to manage and edit issues in the Issues in Advanced Roadmaps section.
- **Teams** - Teams represent the people who will be doing the work on your timeline. Learn more about how to create and assign issues on the Teams in Advanced Roadmaps page.
- **Releases** - Set milestones on your timeline and track your progress against them with Releases. Learn how to create and manage them on the Releases in Advanced Roadmaps page.
- **Capacity** and **velocity** - As you build your plan, you'll want to ensure that you're not over-assigning work to one team. Advanced Roadmaps has a depth of capacity management tools to help you create balanced and actionable plans for your teams based on past averages. Learn how to assign and manage capacity in the Capacity and Velocity section.
- **Dependencies** - A dependency signals order of tasks in your plan. Issues can either "block" or be "blocked by" other tasks on your timeline, indicating that one must be done before the other. Learn how dependencies can help you convey a sense of order in your plan on the Dependencies in Advanced Roadmaps page.
- **Warnings** - Advanced Roadmaps alerts you to any inconsistencies or errors as you create your plan. Some common warnings are misaligned dates, misconfigured assignments, or simple oversights. Learn more about how Advanced Roadmaps helps you create a clean and effective plan in the Warnings in Advanced Roadmaps section.

Lastly, Advanced Roadmaps is a sandbox environment meaning that any changes you make to your plan won't be committed back to your Jira issues until you choose to do so. Learn how to Save your plan to Jira issues once you're ready to put your plan into action.

> ⓘ Advanced Roadmaps is now part of Jira Software Data Center. To get access to the features described on this page, upgrade to Jira Software Data Center 8.15 or later. Can't upgrade yet? Depending on your current Data Center version, you can access these features by installing the latest version of the app (at no cost). See our FAQ for all the details

Plans in Advanced Roadmaps

Simply put, a plan contains all of the work that needs to be done on your project in a timeline format. Every piece of work is broken down into issues assigned to teams. Advanced Roadmaps is a sandbox environment meaning that any changes you make to your plan won't be committed back to your Jira Software issues until you choose to do so.

Create a new plan

> The Plan creation wizard was introduced in the Jira Data Center and Server 8.19 release. If you're using the standalone Advanced Roadmaps app from the Marketplace (version 3.29 or earlier), please refer to the 8.15 version of this page.

To create a new plan in Advanced Roadmaps:

1. From the navigation at the top of the page, go to **Plans** > **Create plan** to open the plan creation tool.
2. Enter these details:
   a. **Plan name** - This will help your team identify the plan.
   b. **Access** - Select **Private** or **Open**. By default, access is set to Open.
   c. **Issue sources** - Choose which projects and boards, and filters you want to import from Jira Software. These will form the plan's scope. Learn more about issue sources.
3. Optional - Select **Set exclusion rules** to refine the displayed issues. This step isn't required, but helps you refine your plan. Keep in mind that there are loading limits for plans in Advanced Roadmaps. You can set rules for:
   a. Completed issues — Enter the number of days completed issues will be displayed in the plan.
   b. Issue types
   c. Issue statuses
   d. Releases
      Use **Back to create plan** in the upper left corner of the modal.
4. When you're satisfied with your plan, select **Create**.

Exclusion rules

You can also exclude issues from your plan based on any of the following rules:

- the date it was completed
- issue type
- issue status, or
- the release to which it's assigned

To set an exclusion rule:

1. From the **Roadmap** view, navigate to settings using the next to your plan name > **Configure**.
2. Select **Exclusion rules** from the menu on the left.
3. Use the fields to tell Advanced Roadmaps which issues to exclude from your plan.

Limits on plan size

To prevent Jira Software from timing out, there are limits on how big your plan can be.

Issue limits

A single plan can load up to 5,000 issues and can't contain more than 100 projects. For larger projects, consider spreading the work across multiple plans.

Team limits

While there is no enforced limit on the number of teams that can be included in your plan, we recommend adding no more than 50 teams. If you add more, some teams won't display on the Teams tab and you'll need to enter the team name in the search box.

Issues in Advanced Roadmaps

Issues are the core building blocks of your plan. You can import issues from issue sources as covered on the Plans in Advanced Roadmaps page, or you can create new issues directly in your plan then save them to your project.

Bear in mind that Advanced Roadmaps is a sandbox environment. Any changes made here, including creating or removing issues, won't affect your Jira Software data unless you tell Advanced Roadmaps to save your changes to Jira.

Create a new issue

To add a new issue:

1. Navigate to **+Create issue** at the top of the **Scope** column.
2. Select the **type of issue** (epic, story, subtask) you want to create. If your plan contains more than one project, use the dropdown to select which project to assign the new issue.
3. Name your new issue, and choose a project if you're making an epic. Select the **checkmark** to save.
4. Add other issue details in the fields displayed on your timeline such as **assignee**, **team**, **estimate**, **sprint**, **release**, and **dates**. The details you can specify for an issue will depend on your issue sources and how your plan is configured.

Advanced Roadmaps will place your new issue at the bottom of your current project. If you've made a child-level issue (such as a story or subtask), it will be listed under the collapsible **x issues without parent** heading at the bottom of your timeline until it's linked to a parent task, which we discuss below.

Insert a new issue

Alternatively, if you know where you'd like to add a new issue, you can tell Advanced Roadmaps where to create a new issue relative to another. This is handy for inserting epics at specific places in projects, or creating a child issue that inherits details from its parent task. On your timeline, hover over the desired destination issue, select the ••• menu, then follow the same steps listed above.

If you insert an epic into your plan using this method, it will be automatically placed in the corresponding project. You can change this using the dropdown menu before selecting the **checkmark** to save it.

If you create a child issue, it will be nested underneath the parent ticket and inherit its **sprints**, **releases**, and **teams** values.

Link issue to a parent

To create a relationship between two issues, drag and drop a child into its parent issue in the **Scope** section. You can only link an issue to a parent of the corresponding hierarchy level that you've configured for your plan.

Schedule issues

How you can schedule issues depends on how your teams are set up, and how you've configured your plan. For example, hierarchy configurations, estimation units, and custom dates all need to be configured before they can be used to schedule issues in your plan.

To learn the different ways that you can schedule issues in your plan, see Schedule issues on your timeline.

Remove issues from your plan

There are two ways to remove issues from your plan in Advanced Roadmaps. The first way is to select the **checkboxes** of the issues you want to remove, then navigate to the **Bulk actions** menu > **Remove from plan**.

Unsaved issues can also be removed using the **Review changes** modal. Select the issues you want to remove, then select **Discard selected changes** to remove them from your plan.

Regardless of how you choose to do it, issues that have been saved to Jira won't be affected when you remove them from your plan.

Rank issues on your timeline

Along with functions like dependencies, ranking issues tells Advanced Roadmaps in which order your work is to be completed. Issues are ranked relative to other issues of the same hierarchy level in your plan. These rankings can't be surfaced in Jira Software, but they're essential for using the Auto-scheduler.

Advanced Roadmaps displays an issue's ranking in the **Scope** column under **#** of the highest hierarchy level displayed in your plan. All issues are ranked relative to those with the same hierarchy level, meaning the ranking of an epic won't impact the ranking of a story.

To see how the child issues are ranked relative to issues in the same hierarchy, limit the hierarchy range visible on your timeline.

To change the ranking of one issue in Advanced Roadmaps, drag and drop an issue to a new row in the **Scope** section. Its child issues will move with it to the new row on your plan, but their ranking won't be changed in Jira Software. If you want to re-rank multiple issues at one time, see Edit issues in bulk from your Advanced Roadmaps timeline.

Any changes you make won't be reflected in your Jira instance until you save them to Jira.

Schedule issues on your timeline

An important aspect of planning work is scheduling how long each issue will take to complete. How you schedule issues depends on how your teams are set up, and how you've configured your instance. For example, hierarchy configurations, estimation units, and custom dates all must be configured before they can be used to schedule issues in your plan.

Advanced Roadmaps can also assist you in scheduling issues using features like the Auto-scheduler and rolling up values to parent issues.

Schedule using start and end dates

The easiest way to schedule an issue is to add a value to the **start date** and **end date** fields on your timeline. Once you've scheduled child issues, you can then use the Rolling-up values to parent issues feature to extrapolate dates of the parent issues.

Alternatively, use your mouse to place a schedule bar on your timeline which you can then expand or contract to change the issue's duration.

Schedule issues according to sprints

> (i) This only applies to Scrum teams.

Once configured by your administrator, you can tell Advanced Roadmaps to schedule issues to sprints provided the issues have a sprint and team assignment but no due dates. When Advanced Roadmaps uses sprint dates, the start and end date columns in your plan will display an S lozenge.

From here, you can then change an issue's dates manually without changing its sprint assignment. However, if the new dates fall before or after the sprint, you'll see a warning.

When a sprint is completed, it will appear as finished on the last day. This prevents sprints from overlapping on the timeline and applies whether your issues are scheduled manually or through sprint assignment.

External sprints

If you use projects or filters as your issue sources, their sprint assignments will still show on your plan. However, since sprint data is based on boards, Advanced Roadmaps will add the EXTERNAL lozenge to indicate that these sprints are not directly associated with the project and filter issue sources. You can update the sprint values of the issues assigned to these external sprints. However, you can't assign new issues in your plan to an external sprint.

To avoid creating external sprints, we recommend that you use boards as issue sources where possible as a board only covers one sprint. Issues imported from projects and filters may comprise multiple sprint assignments which will show on your timeline as external sprints.

Troubleshoot external sprints

There are two main reasons you may see external sprints on your plan:

- One of your future sprints appears on another Scrum board
- One of the teams for which you're planning future sprints has a different iteration length configured

If either or both of these are true, and you use **Group issues by team** and **Show capacity on the timeline** to plan future sprints based on capacity and velocity, the common sprint will occur at different times in each team swimlane. To compensate for this inconsistency in planning, Advanced Roadmaps will automatically create an external sprint because the same sprint can't have two different start and end dates.

This issue will resolve itself once a future sprint becomes an active one.

External teams

When an issue is assigned to a team that's not part of your plan, it shows as an external team. To unlock the full planning capabilities for these teams in your plan, you first must add them using the Teams in Advanced Roadmaps tab.

Auto-schedule issues in Advanced Roadmaps

The Auto-scheduler constructs a plan using issue details based on your plan settings. You can then adjust the generated plan to meet your exact needs. Advanced Roadmaps considers several factors to come up with the ideal schedule for your teams:

- whether your teams work in Scrum or Kanban method
- the sprint assignments of the issues in the plan (if applicable)
- remaining capacity of the iteration
- start and end dates of issues
- issue rankings
- dependencies
- issue estimates
- the number of members in your team

To auto-schedule issues in your plan:

1. Navigate to **Auto-schedule** and configure the auto-schedule settings.
2. Select **Preview results** to see suggested changes, as shown above. The schedule bars of auto-scheduled issues and their corresponding issue details will be shown with purple stripes. Hover on a field to compare the current values against the suggested changes. The number of auto-scheduled issues will depend on the hierarchy levels you've chosen for the plan.
3. Select **Accept changes** to save the changes. These changes are only applied to your plan and are not yet saved in Jira Software. You'll need to use **Review changes** to save changes to the Jira Software issues.

You'll need to accept or decline the changes made by the Auto-scheduler before you can use the **Review changes** function. The Auto-scheduler will not overwrite any issues that are in a sprint that's currently active.

Auto-schedule for Scrum teams

When working with Scrum teams, the Auto-scheduler will not change or assign sprint values for issues at the epic level and higher. To avoid duplicating values and issues, we recommend that you only estimate issues at the story level, and that you show rolled-up values in your plan.

For Scrum teams, the Auto-scheduler will assign tasks to sprints based on the size of an issue, team capacity, and sprint length as configured in your plan settings. Any excess issues will be assigned to the next sprint on the timeline. If an epic spans multiple sprints, work will be allocated between them based on available capacity. If the work extends beyond currently existing sprints, Advanced Roadmaps will assign future work to projected sprints.

Auto-schedule for Kanban teams

For Kanban teams, issues are scheduled on a per day basis based on the team's weekly capacity. Advanced Roadmaps assumes a weekly capacity of 200 hours, and the iteration length cannot be changed.

Auto-schedule capacity

The Auto-scheduler will schedule issues based on your plan's capacity, velocity, and iteration length. If issues are scheduled for projected iterations that don't yet exist in your Jira Software instance, their capacity may not be accurately reflected in your plan until a real sprint is created in the board backlog.

Learn more about how capacity is used in Advanced Roadmaps.

Roll up values to parent issues on your timeline

> (i) The ability to roll up values to parent issues was introduced in Advanced Roadmaps version 3.23. If you're using an earlier version of Advanced Roadmaps, this process may be different or not available in your instance.

Once you set values for child issues, you can tell Advanced Roadmaps to apply those values to parent issues as well. For example, if you roll up the dates on a story with three scheduled subtasks, Advanced Roadmaps will set the values of the parent issue according to the earliest start date and latest end date of the child issues. These auto-estimated values will appear in the parent issue next to a roll-up arrow icon (shown below), and reflect any changes to child issues.

In Advanced Roadmaps, you can roll up the following values:

- estimates
- teams
- sprints
- releases

Your rollup values are effected based on filters applied to your plan. Issues that are filtered out won't be included.

Roll up estimates in a plan

When issue estimates are rolled up from child issues to parents, values are rounded to one decimal place for story points, and two decimal places for hours and days. The rolled-up estimate of a parent issue will reflect how much work is yet to be done, and will decrease as you log time to child issues.

Roll up dates in a plan

Roll-up dates are treated the same as manually-set dates in Advanced Roadmaps. By default, Advanced Roadmaps rolls up target dates, but you can configure your plan to use custom dates or sprint dates. Rolled-up dates will also respect the group and color settings of your plan.

Any explicitly set dates will override rolled-up dates. Dates of child issues can only be rolled-up if the parent issue does not have an assigned start or end date and is not part of a sprint.

In order to roll up dates, a parent issue must have at least one child issue with start and end dates assigned. Given that, Advanced Roadmaps will fill the missing date values based on the child issues' dates.

Partial roll-ups

If your schedule bars have a striped pattern that fades out in one direction, this indicates that a parent issue is using one set date and one date rolled-up from child issues. This is called a partial roll-up, and is demonstrated below:

If a parent issue's schedule bar fades out completely, it means that none of its child issues have a start and end date assigned to them. Missing start or end dates prevent Advanced Roadmaps from determining a roll-up date, and so the schedule bar fades out.

To fix a partial roll-up, ensure that the parent issue contains at least one child issue with start and end dates.

Edit issues in bulk from your Advanced Roadmaps timeline

Advanced Roadmaps lets you change values of multiple issues at the same time using the **Bulk actions** menu.

Select issues

To begin, select the issues you want to edit. Use the checkboxes in the **Scope** column to select individual issues or hold **Shift** on your keyboard to select a range of issues at once. This method will only select the issues displayed in your plan, and not collapsed child or descendant issues. To select them, expand the parent issue before selecting.

Alternatively, if you would like to select an entire story or epic, go to the **more** ••• icon in the **Issues** column. You can choose to either **select this and all descendant issues** or **select only descendant issues**.

The **Bulk actions** menu will only appear after you've selected an issue.

Reassign issues to individuals, teams, or sprints

Once you've selected your issues:

1. Select **Assignee**, **Team**, or **Sprint** from the **Bulk actions** menu.
2. Use the dropdown menu to reassign the issue. **Unassigned** or **None** will clear the assignment field.
3. Select **Apply** to complete your changes.

Update issue dates

Before you can change issue dates in bulk, you must first associate the custom date field with the issue sources of the plan. Once selected, the custom dates will be displayed with the date lozenge (D).

Once you've selected your issues:

1. Select **Dates** from the **Bulk actions** menu.
2. Choose to **Keep existing**, **Clear date**, **Set by child issue dates** which will aggregate the dates of child issues up to the selected parent issues, or **Set custom date**.
3. Select **Apply** to complete your changes.

Update parent issues

In order to move multiple child issues, they must be of the same hierarchy level and cannot be the highest level in your plan (for example, epic).

To update parent issues:

1. Select **Parent** from the **Bulk actions** menu and choose the parent issue to which you'd like to move the selected child issues. To clear the current values for the selected issues, select **None**.
2. Select **Apply** to complete your changes.

Update issue rank

Once you've selected the issues you'd like to re-rank in the **Scope** section, there are two ways to move them. First you can navigate to the **Bulk actions** menu > **Rank**. Choose the new rank for the selected issues from the following options:

- **Top of plan** - rank issues as highest priority and move to the top of the plan
- **Bottom of plan** - rank issues as lowest priority and move to the bottom of the plan
- **Above issue** - rank issues above a specific issue of your choosing
- **Below issue** - rank issues below a specific issue of your choosing

Select **Apply** to move the issues to their new rank.

The second way is to select the issues you'd like to re-rank. Hover over the issue where you would like to place them in your plan, and select **more** ••• > **Rank selected issues above/below**.

> ⓘ Re-ranked issues may appear differently than their actual rank because they will be visually ordered according to the parent-child relationships in Advanced Roadmaps. Their new rank will be more accurately displayed in Jira Software.

Update issue releases

The process of updating release details for issues is the same whether they are in the same project or part of a cross-project release.

Once you've selected your issues:

1. Navigate to the **Bulk actions** menu then select **Release**.
2. In the modal, choose the destination release into which you would like to move the selected issues. Only releases that are associated with projects of the selected issues will be available. Choose **None** to clear the current values for the selected issues.
3. Select **Apply** to complete your changes.

Edit Jira Software issues from Advanced Roadmaps

Select **Bulk change in Jira** in the **Bulk actions** menu to make changes directly to issues in Jira Software. You can edit, move, transition, delete, watch, and stop watching issues while in the context of a plan.

Remove issues from a plan

To delete issues from your plan, go to **Remove from plan** in the **Bulk actions** menu and confirm your action by selecting **Remove**. Issues that aren't saved in Jira Software will be permanently deleted.

Troubleshoot missing issues in Advanced Roadmaps

There are a few reasons why issues in your plan might not show as expected. This page explains the most common reasons for missing issues and shows you how to display them.

Resolution field isn't set in the Jira Software workflow

When you're overseeing the progress of large, longer-term projects, it can be helpful to track completed issues and keep them displayed in your plan. If completed issues are missing from your plan, check the Resolution field in your Jira Software workflow.

Jira Software uses the Resolution field to define completed issues. When you're using a default workflow, the Resolution field is automatically set when an issue is transitioned from the "In Progress" to "Done" status. This is due to a post function on the incoming transition to "Done" that's configured to set the field.

If you're using a custom workflow, this needs to be manually configured. Jira Software admins can add a post function on the incoming transition for the status being used to signal a "Done" issue. Without this post function, completed issues in a custom workflow won't be displayed in your plan.

Learn how to configure the Resolution field as a post function in your Jira Software workflow.

Issue was completed more than 30 days ago

By default, Advanced Roadmaps doesn't display issues completed in Jira Software more than 30 days ago. You can adjust this timeframe to suit your planning needs — whether you need to keep your plan focused on specifics, or to track completed work as part of the bigger picture.

To adjust how long Advanced Roadmaps should display issues after they're completed:

1. From the Roadmap tab of your plan, go to ⚙ > **Configure**.
2. Select **Issue sources**.
3. Under **Completed issues**, enter the number of days you'd like completed issues to be displayed in your plan.

The more completed issues you include in your plan, the longer it'll take to load.

Issue isn't included in the plan's issue sources

Issues in your plan are determined by the issue sources selected when the plan was first created. Any issue source that's not included is considered out of scope, and won't be displayed in the plan. You'll need to make sure the missing issue is included in an issue source for the plan's scope.

To add an issue source to your plan:

1. From the Roadmap tab of your plan, go to ⚙ > **Configure**.
2. Select **Issue sources** > **Edit issue sources**.
3. You'll be taken to the setup wizard. On the **Connect to your teams' work** page, you can add an issue source at the bottom of the list.
4. Select a **board**, **project**, or **filter**.
5. Select the **issue source** you'd like to add to your plan. You can click to select or start typing to narrow the selection.
6. Select **Next** > **Next** to continue through the rest of the wizard.
7. Select **Done**.

Issue's release was excluded in the plan's scope

Similar to issue sources, releases selected during plan creation determine if an issue is part of a plan's scope. Any excluded releases and the issues assigned to them won't be displayed in the plan.

To add a release to your plan:

1. From the Roadmap tab of your plan, go to ⚙ > **Configure**.
2. Select **Issue sources** > **Edit issue sources**.
3. Select **Next** to go to the second page of the setup wizard where you can **Select relevant releases**.
4. Select the missing issue's release, then select **Next**.
5. Select **Done**.

Issue was manually excluded from the plan

Depending on how your plan was first set up, specific issues may have been manually excluded. When issues are manually excluded, they won't be displayed in the plan regardless of their source or release.

To re-include an issue in your plan:

1. From the Roadmap tab of your plan, go to ⚙ > **Configure**.
2. Select **Issue sources** > **Edit issue sources**.
3. Select **Next** > **Next** to reach the third page of the setup wizard where you can **Confirm what's in scope.**
4. Use the **Projects** and **Sources** dropdowns to filter your search of the missing issue. You can also expand and collapse issue hierarchies in the list.
5. Once you've found the missing issue, select its checkbox to include it in the plan.
6. Select **Done**.

Fix version field is hidden in Jira Software issues

Releases in Advanced Roadmaps are known as Fix versions in Jira Software. If the Fix version field is hidden in your Jira Software issues, Advanced Roadmaps won't associate the issue with a release. This could cause missing issues if you've used releases to form the scope of your plan.

To display the Fix version field in Jira Software:

1. Navigate to the Jira Software admin panel using the ⚙ icon in the upper right and choose **Issues**.
2. Select **Field configurations** from the menu on the left.
3. Find the relevant field configuration for your plan and select **Configure**.
4. Go to the **Fix versions** section, then select **Show**.

Teams in Advanced Roadmaps

> ℹ️ The **teams** field is a custom field in Jira Software. To make this field appear in Jira Software issues, you'll first need to add this custom field to the screen that's being used in the corresponding Jira Software project.

Once configured, Teams in Advanced Roadmaps act as a label to designate which team will eventually pick up the work on your timeline. Once saved, you can search for issues that have been labeled for that team, then assign them to individuals when grooming your backlog.

There are two types of teams to choose from in Advanced Roadmaps:

- Private teams only exist within your plan so you can explore what-if scenarios and see how changes to your teams will affect your plan. This is what Advanced Roadmaps refers to as Scenario planning.
- Shared teams are visible across your organization and can be assigned to work on multiple plans.

In the **Teams** tab underneath your plan name, you'll see the list of teams included in your plan, as in the example below:

This team snapshot includes:

- the issue source associated to the team
- the capacity and iteration length (capacity information will vary depending on whether it's a Scrum or Kanban team, as well as how their work is estimated)
- members of the team
- whether the team is shared or private (indicated by the lozenge)

You can also **Filter issues by this team** to view only the issues assigned to the specific team in your plan.

> ℹ️ You need Advanced Roadmaps user permission to save team changes in Jira Software.

Create a new team

Every team you create in Advanced Roadmaps will begin as a private team and will only exist within this specific plan. Once created, you can convert a private team to a shared team. This cannot be reversed.

To create a new private team:

1. In the **Teams** tab, navigate to **+ Add team** > **New private team**.
2. Choose a **name** for your team, then select an issue source that you want to associate with the team. Issues from this issue source will be given a higher priority for this team. Learn more about Issue sources.
3. Choose whether your new team will be a Scrum team or a Kanban team, then set the **iteration length** and **capacity**. The type of team you choose will have an impact on how capacity is handled and managed in a plan.
4. Add **members** to the team.
5. When you're happy with your team, select **Create**.

Your team will now be available in your plan, but the changes won't be finalized until you Save selected changes in Jira using the **Review changes** button.

ℹ️

Convert a plan-only team to a shared team

> ⓘ This process can't be reversed.

To convert a plan-only team to a shared team:

1. Navigate to the **Teams** tab
2. On the team you'd like to convert, open the ••• menu, then choose **Share**
   a. If the **Share** button doesn't show (and your team has a yellow bar on its side), you'll need to save the team first using the **Review changes** button. Learn more about reviewing and saving changes in Advanced Roadmaps
3. Select **Share** in the modal that pops up.

Once shared, you can then assign it to issues across multiple plans.

Add a shared team

To add a shared team to a plan:

1. In the **Teams** tab, navigate to **+ Create team** > **Add existing shared team**.
2. Start typing to find the team you'd like to add. Select the team > **Add shared team**.

Edit a team

Whether it's a private team or a shared team, you can edit the following details:

- team name
- members
- associated issue source
- capacity and iteration length
- team type (only if all teams in the plan use time-based estimates)

To edit a team, navigate to the **Teams** tab and select **more** ••• > **Edit** for the team you want to edit. Modify the details of the team, then select **Save**.

Assign work to a team

Once your teams are set up, you can easily assign any issues by typing the team's name into the **Team** column of your plan.

Delete a team

When a team is no longer needed, you can remove them from your plan. Shared teams can be re-added to your plan, but you can't recover a deleted private team.

To delete a team

1. Navigate to the **Teams** tab
2. Select **more** ••• > **Delete**, and Advanced Roadmaps will remove the team from your plan.

Capacity and velocity in Advanced Roadmaps

> (i) Capacity and velocity planning was introduced in Advanced Roadmaps version 3.28. If you're using an earlier version of Advanced Roadmaps, this process may be different or not available in your instance.

The number of story points or hours your team can complete in one iteration is referred to as its capacity. For example, an iteration with a capacity of 30 story points (which is the default setting) can contain six issues that are estimated at five story points each. You can use capacity to plan out different timeline possibilities, referred to as Scenarios in Advanced Roadmaps. Advanced Roadmaps handles capacity differently depending on a number of settings including how your teams estimate work, whether they're a Scrum or Kanban team, and more.

Velocity, which is used strictly for planning purposes, is your team's average capacity per iteration over time based on past performance. For example, a team with a velocity of 30 story points will require two sprints to complete an epic estimated at 35 story points, assuming no other work was already assigned to those sprints.

Enable capacity management in your timeline

To view capacity on your timeline:

1. In the **View settings** menu, select **Team** or **Sprint** from the **Group by** menu. This will organize your issues into swimlanes.
2. Select the checkbox next to **Show capacity on timeline** to finish.

To manage a team's capacity from your timeline, you must use a board with sprints or iterations as the issue source for your plan. For scrum teams, the teams in your plan must also be associated with that board.

Calculate capacity for Scrum teams

When scheduling work for Scrum teams, capacity is calculated differently depending on which units you use to estimate work:

- If you're using time-based estimates, the sprint capacity is determined by taking the weekly capacity and multiplying by the number of weeks per iteration. For example, if you set the capacity at 40 hours, the sprint's capacity will be 80 hours, assuming a two-week sprint.
- If you estimate in story points, no conversion is necessary since story points are defined per iteration. The capacity applies to the whole sprint. By default, this is set to 30 story points, though you can adjust this to meet your team's needs.

Calculate capacity for Kanban teams

> (i) Kanban teams can only use time-based estimates.

For Kanban teams, work is broken down into iterations which each last one week. You can set the weekly capacity of a Kanban team in either days or hours, but the length of the iteration cannot be changed.

Assign capacity from your timeline

> (i) Capacity and velocity planning was introduced in Advanced Roadmaps version 3.28. If you're using an earlier version of Advanced Roadmaps, this process may be different or not available in your instance.

When an issue is added to an iteration, it will consume the estimated value (story points or days/hours) from the capacity value set by your administrator. Issues that span multiple iterations will consume all remaining capacity of the first iteration before moving on to the next one. Capacity is assigned on a first come, first served basis meaning that pre-existing issues won't be moved. This is the same whether you're planning in story points or time-based estimates.

Assigned issues that consume more capacity than is available in an iteration will generate a warning. We recommend that any assigned issues that have an estimation value higher than the capacity be broken down into smaller subtasks or spread across multiple iterations.

> (i) You won't see warnings if they're not enabled in your timeline. To turn them on, see Warnings in Advanced Roadmaps.

In order for the capacity function to work properly, all issues assigned to a sprint must be estimated. If you include an issue that hasn't been estimated, it won't consume capacity. For this reason, epics can span more than one sprint without consuming capacity or generating a warning.

Issues that are assigned start and end dates that fall within a sprint will consume capacity, even though they're not assigned to that sprint. When monitoring capacity, these issues are marked with the label **x issues not assigned to sprint**.

Change sprint capacity

The capacity of your sprints can fluctuate for several reasons. Examples include public holidays, a sick team member, or team members spread across different projects.

In these situations, you can keep the sprint (and therefore your plan) accurate by editing its capacity using the box next to the **Sprint velocity** field in the sprint details flyout:

When you're done, select the **checkmark** to confirm the change. This can be done for active and future sprints.

Scenarios in Advanced Roadmaps

The process of planning large pieces of work can involve assumptions, and these assumptions can cause a ripple effect in your plan. The **Scenarios** tool in Advanced Roadmaps helps you to plan for and assess the impact of possible outcomes by mapping out different "what if" versions of your plan. When using this tool, any changes you make to issues will only be applied to the currently selected scenario.

These modified versions of your plan only live in Advanced Roadmaps unless you save the changes to Jira Software.

Enable scenarios

To enable scenarios in your plan, go to **Settings** ⚙ > **Configure** > **Scenarios**, then select **Enable scenarios**. If you skip this step, you won't see Scenarios in your plan.

Create a scenario

Once enabled, you can create scenarios directly from the Roadmap tab.

To create a scenario, navigate to the **Scenario** menu next to **Settings** ⚙ > **Create scenario**. Give your scenario a name and choose a color for it. Any changes you make in this scenario will be displayed with flags in the color you select.

When creating a new scenario, Advanced Roadmaps gives you two options for starting points:

- A **blank scenario** will create a new scenario based on the data from your Jira Software issues. It will not include any unsaved changes in your Advanced Roadmaps plan.
- A **copy of** an existing scenario will inherit changes from the scenario of your choice.

When you're done, select **Create**.

Once you've created multiple scenarios, you can switch between them using the **Scenario** dropdown menu.

Save a scenario to Jira Software

Advanced Roadmaps is a sandbox environment which means that any changes you make won't be saved to Jira unless you choose to save them. Once you've created and modified your scenarios, select the appropriate scenario for your team, then save it using the **Review changes** button. Doing so will overwrite your data in Jira Software with these new values.

Manage scenarios

Make changes to your scenarios by navigating to the **Scenario** menu > **Manage scenarios**. In this menu, you can do the following:

- Update the name of a scenario by selecting the **name**.
- Change the assigned color of the scenario by selecting the color swatch.
- Delete a scenario using the **delete** icon.

Alternatively, you can access the Scenarios page by going to **Settings** ⚙ > **Configure** > **Scenarios**.

Turn scenarios off

If you choose to disable scenarios in your plan, you'll be able to keep the changes of only one scenario in your plan. All other scenarios will be lost.

To disable scenarios, navigate to **Settings** ⚙ > **Configure** > **Scenarios** and select **Disable**. From the dropdown menu, select which scenario you'd like to save in your plan, then confirm your choice by selecting **Dis able**.

Dependencies in Advanced Roadmaps

> (i) Your plan must be configured to use issue links in Jira Software, and the relationships between issues will need to be defined in a way that's suitable for your team. For more information, see our page on Configuring plan settings.

Dependencies in Advanced Roadmaps indicate which issues are contingent on others being completed first. On your timeline, dependent issues will have numbered badges at either end of the schedule bar based on dependency type. There are two types of dependencies in Advanced Roadmaps:

- Incoming dependency ("is blocked by") means the issue is dependent on the previous issue being completed, indicated by a badge on the left side of the schedule bar.
- Outgoing dependency ("blocks") means the issue blocks the next issue being started, indicated by a badge on the right side of the schedule bar.

In the picture below, ADR-24 has an outgoing dependency while ADR-23 has an incoming one. These badges are referring to the same dependent relationship, only from different perspectives.

Learn how to monitor dependencies on your timeline.

Add a dependency

To add a dependency to an issue in your plan:

1. Hover on the schedule bar for the issue to which you want to add a dependency, and select the **+ icon**. Selecting the + icon on the right will create an outgoing dependency, while the left will create an incoming dependency.
2. Choose the issue to which you want to attach the dependency.
3. Select the **checkmark** to confirm your choice.

After adding a dependency to an issue, adjust the dates by clicking and dragging the ends of the schedule bar. The badges on either end will turn red if there are conflicting dates.

Warnings and scheduling of dependencies are based on whether your plan is configured to handle dependencies sequentially or concurrently.

Remove dependencies

Removing dependencies from issues will only dissolve the relationship between the issues. It won't remove the issues from the plan.

To remove dependencies from an issue:

1. Display an issue's dependency details by hovering over the dependency badge.
2. Select the **delete** icon next to the dependency.

471

Releases in Advanced Roadmaps

> ⓘ Releases in Advanced Roadmaps are referred to as Fix versions in Jira Software.

> ⓘ You need the Advanced Roadmaps **user** permission to save changes in Jira Software.
>
> Additionally, you will need the **Administer Projects** permission in all projects associated with the cross-project release.

You can track projects with deadlines in mind using the **Release** functions in Advanced Roadmaps. Each release can contain the work of one or multiple teams to give you a higher-level view of your goals. Once you associate a project with a release, you can then track their progress against your projected due date.

Viewing and managing releases is all done in the **Releases** tab underneath your plan name. This list will include any set releases (or fix versions) that you've created in Jira Software in your plan, as shown below:

Before you begin scheduling against a release date, review your teams' capacity settings to ensure that your target date is achievable. These settings will impact how tasks in sequence are scheduled. Once you confirm these settings, you can map out releases and schedule them in your plan.

This page covers setting up and tracking work against releases in Advanced Roadmaps. To learn how to monitor the health of your releases, see our Monitor releases from your timeline page.

Release types: Project release versus cross-project release

There are two types of releases which are treated slightly different in Advanced Roadmaps. A single-project release only contains one project. A cross-project release combines multiple project releases into one plan so you can align dates, prioritize work, and monitor multiple projects at once.

One key difference is that a single-project release can be saved in Jira Software as a fix version. A cross-project release is only used for planning purposes in Advanced Roadmaps. When you save a cross-project release to Jira Software, only the single-project release associated with the teams in your plan will be saved.

Create a single-project release

1. From the **Releases** tab, select **Create release** next to the project name.
2. Name your release and add a description.
3. Choose a **start date** for this release. Advanced Roadmaps gives you the following options:
   a. **As early as possible**: When auto-scheduling the plan, the release will start as soon as there are available teams to work on the issues in the release.
   b. **Fixed date**: You can set a preferred date for the start date.
   c. **Relative to previous release date**: You can specify the gap of time that you want to allocate after the previous release finishes, and work on the new release starts. For example, you can start work on the beta release one day after work for the alpha is completed.
4. Choose a **release date**. Advanced Roadmaps gives you the following options:
   a. **After all issues are completed**: The release date will be automatically generated based on when the issues in the release are scheduled to be completed using predefined capacity and velocity.
   b. **Fixed date**: The release will happen on this chosen date, overriding the capacity and velocity settings. The release icons on the timeline will turn red for overbooked releases. See Monitoring releases for more information.
5. When you're happy with your release, select **Create** and your new release will be added to the plan.

Your release will now be available in your plan, but the changes won't be finalized until you Save changes made on your timeline to Jira Software using the **Review changes** button.

Create a cross-project release

1. From the **Releases** tab, select **Create release** next to the project name.
2. Name your release, add a description, and choose the **project** and **release** you want to add. Then select **Add release**.
3. Repeat to add as many releases as necessary.
4. When you're happy with your release, select **Create** and your new release will be added to the plan.

> ⓘ Cross-project releases are only used in Advanced Roadmaps and cannot be saved to Jira Software.

Edit a release

You can edit the releases in your plan at any given time.

1. Start by navigating to the **Releases** tab and selecting more ••• > Edit on the release you'd like to edit.
2. Make the changes you want and then select **Save changes** to update the release details in your plan. The changes won't be finalized until you Save selected changes in Jira Software using the **Review changes** button.

Align release dates in cross-project releases

When managing work across multiple releases, keep an eye out for misaligned release dates. You can quickly fix these by aligning the release dates in a cross-project release.

To align the dates in your cross-project release:

1. Navigate to the **Releases** tab and find the cross-project release whose release dates you want to align.
2. Select **more** ••• > **Align dates** which will show you the same date options discussed when creating a single release.
3. When you're ready, select **Set dates** to update the dates in your plan.

Remove and delete releases

Over time, releases that have already been shipped or completed will start to accumulate in your plan. Whether you can remove or delete a release depends on if the release has been saved to Jira Software. If the release has been saved in JJira Software, you can remove it from your Advanced Roadmaps plan though it will remain in your Jira instance. If your release has not been saved to Jira Software (either because you haven't saved it yet or because it's a cross-project release which can't be saved) you'll get the option to delete the release. Once deleted, releases cannot be recovered.

To remove a project release:

1. In the **Releases** tab, find the single-project release you want to remove from your plan.
2. Select **more** ••• > **Remove**. Then select **Remove** in the modal window to confirm.

To delete a release:

1. In the **Releases** tab, find the release you want to delete and select **more** ••• > **Delete**.
2. On the modal, you'll see the option to **Remove all releases from the plan**. This will not only delete your cross-project release. It will also remove the included releases from your plan. Any project releases that have been previously saved to Jira Software will still exist in Jira. Any releases that haven't been saved will be permanently deleted.
3. When you're ready, select **Delete** to confirm.

Warnings in Advanced Roadmaps

While scheduling work for your teams, Advanced Roadmaps will alert you to errors such as misaligned dates in your plan. Using the warnings dropdown menu, you can modify which warnings are shown, filter issues by warning type, and troubleshoot common warnings.

This page discusses the warnings that you'll find in your plan. If you see warnings in the Review changes modal, see Save changes made on your timeline to Jira Software to resolve them.

Enable warnings

In order to show warnings in the dropdown menu, navigate to the **View settings** menu and select the **Show warnings** checkbox. Any warnings in your plan will be marked by an icon in the Scope column next to the affected issue, as shown below:

You'll also see the total number of warnings displayed at the top of your plan:

Select the **x warnings** menu to view your warnings.

View the warning center

> ⓘ The warning center was introduced in Advanced Roadmaps version 3.29. If you're using an earlier version of Advanced Roadmaps, this process may be different or not available in your instance.

Once you open the warnings menu, you can manage them in several ways:

1. Toggle the switch labeled **Filter issues with warnings** to only show issues that have warnings associated with them in your plan.
2. Select the message in the **Status** column to view all issues with the same warning type. From that list, you can navigate directly to that issue in your plan by selecting the issue name.
3. Toggle the corresponding **Enabled** button to exclude all issues with the same warning type. This will exclude them from your total warning count and hide them from your view:

The "Issue dates outside of assigned sprint" warning is shown as affecting four issues.

Now that warning has been disabled and removed from the total warnings count.

Warning types

For more information about what each warning means and how to fix it, head over to our Troubleshoot warnings in Advanced Roadmaps page.

Troubleshoot warnings in Advanced Roadmaps

As you schedule work for your teams, Advanced Roadmaps will warn you if there are any misaligned dates in your plan. This page discusses common causes of warnings, and how to resolve them.

Jump to:

- Starts before parent
- Target dates set beyond the due date
- This issue has been added to an active sprint
- Unresolved issue after target end date

Starts before parent

If you set dates for issues that don't align with the dates of their parent, Advanced Roadmaps will generate a warning highlighting the conflict, as shown below:

To resolve this, adjust your dates so that the child issues fall within those of the parent. This also applies when using sprint dates in the plan. Read more about how to schedule issues.

Target dates set beyond the due date

Depending on how your plan is configured, your plan view may also show Due dates that Advanced Roadmaps imports from Jira Software. If you set a target end date that extends beyond the assigned due date, Advanced Roadmaps will generate a warning, as shown below:

To resolve this, adjust the target end date or the due date.

This issue has been added to an active sprint

This warning only appears in the **Review changes** modal.

If you add or remove an issue from an active sprint, you'll see a warning in the **Review changes** modal informing you that this will change the scope of the sprint. There's no resolving this issue as it won't break your plan, but it's a heads up that sprint capacity will be affected.

Unresolved issue after target end date

Advanced Roadmaps will warn you when an issue in your timeline has already passed its end date.

To resolve this, update the status of the issue or its end date. If the unresolved issue is in progress, you'll be prompted to do one of the following:

- Move the issue status back to TO DO
- Add the issue to a different sprint
- Reschedule the issue to a later end date

Save changes made on your timeline to Jira Software

> ⓘ To review the changes made in your plan and save these changes in Jira Software, you'll need the Advanced Roadmaps administrator permission. See Permissions for more details.

Advanced Roadmaps is a sandbox environment. Changes you make to values such as estimations, dependencies, due dates, and so on won't be applied to the issues in Jira Software until you choose to do so using the **Review changes** button. If you save your plan while viewing a scenario, that scenario's settings will be committed back to your plan.

Selecting **Review changes** brings up a modal that lists all of the changes made to your plan since it was last saved. From here, you can find specific changes, save them to Jira Software, or discard them.

Save changes

In the **Review changes** modal, select the issues you want to save. By default, all changes will be selected. Any deselected changes won't be saved to Jira Software, though they will remain in your plan. When you're ready, use the **Save selected changes in Jira** button to save your changes.

Filter changes

Sort through changes by using the **Filters** menu in the upper left corner. You can also view change details by expanding the dropdown in the **What changed** column.

Discard changes

In the **Review changes** modal, select the issues you want to discard. These changes will be removed from your plan and the values will revert to those currently saved in Jira Software. When you're ready, use the **Discard selected changes** button to erase them.

Abandon save

While saving or discarding your changes, use the **Quit** button that appears while Advanced Roadmaps is processing your change to stop this action. It won't reverse any changes that have already been saved before you stopped it, but remaining changes won't be saved in Jira Software.

Warnings when saving changes

In the **Review changes** modal, you may see warning icons in the far left column. We recommend resolving these issues before saving your changes to Jira Software. Learn more about warnings here.

Required custom fields in Jira Software issues

If your Jira Software issues have required custom fields, but these fields weren't added to Advanced Roadmaps, you'll see a warning. Since these fields are required, you'll need to either add them to your Advanced Roadmaps plan or turn off the required field in Jira Software.

Custom fields not supported for issues types or projects in plan

If you set a value for a custom field that hasn't yet been set up in your plan, you won't be able to save this change in Jira Software. If you need this change to be saved, ask your Jira Software admin to configure this custom field in your instance.

Child issues not saved to Jira Software

If you've created child issues in your plan, the parent issue needs to be saved in Jira Software as well. Since the child issue is dependent on values inherited from the parent, the child issue can't be saved on its own.

View your Advanced Roadmaps plan

This diagram has features grouped together with each number dedicated to a specific type of action:

1. Manage your view
2. Modify your view
3. Issues in your plan
4. Issue values in columns
5. Timeline schedule bars, dependencies, and releases
6. Clean up your plan and save changes

1. Manage your view

The tabs in the upper left corner of your plan are where you can switch which aspect of your plan you're seeing, and it contains four tabs:

- Your **Roadmap views** are where your project plans come to life. On this screen, you'll see the issues included in your plan laid out with schedule bars to represent their estimated duration. Learn more about creating a plan on the Plans in Advanced Roadmaps page.
- **Teams** is where you'll find more information about the teams included in your plan. As you build your plan, you can assign work to these teams or individuals. Learn more about Teams in Advanced Roadmaps.
- **Releases** is where you configure and manage the releases associated with your plan. Learn how to use releases in your plan.
- The **Dependencies Report** is a visual map of issues that are linked and must be completed in a specific order. Find out more about the Dependencies report.

The ⚙ next to your plan name will take you to the Plan settings menu where where you can configure settings specific to your plan. Learn how to change your plan settings.

The menu on the right is the **Scenarios** menu. Use this to navigate between different versions of your plan, if scenarios are enabled in your plan. If you're not, then this menu won't be here. Learn more about Scenarios in Advanced Roadmaps in Advanced Roadmaps.

2. Modify your view

This area of Advanced Roadmaps is where you apply rules to limit what information is shown on your timeline.

Starting in the upper right, Advanced Roadmaps comes with four preconfigured views and also allows you to create your own custom views that you can recall at any time. Switch between these views, as well as any custom saved views, using the **Views** menu. Learn more about these views on the Preconfigured timeline views in Advanced Roadmaps and Modify your views pages.

You can filter and hide issues from your plan using the level beneath the tabs and the **Views** menu. On the left side underneath the roadmap tab, select which hierarchy levels you want to show on your plan iwth the **Hierarchy** menus, or filter by other criteria using the **Filters** menu. Learn more about how to filter issues on your timeline.

The right side of the header is where you can adjust your view settings. You can choose the length of time you want your roadmap to cover using the timeframe dropdown menu. Choose from 3 months (**3M**), one year (**1Y**), or **Fit** which will adjust the timeframe to fit all scheduled work into view.Advanced Roadmapsalso gives you the option to set your own timeframe. When you select **Custom**, you can choose to either set **Fixed** dates to show a specific range, or **Relative** to show a rolling period of time that is relative to the current date (for example, the past 2 weeks, the next 3 months).

The **View settings** menu lets you customize how issues and related information are displayed on your timeline. In this menu, you can do the following:

- change how issues are sorted
- change how issues are grouped
- add color to issues in your plan
- enable or disable warnings
- turn rollup values on and off and
- enable capacity management on your timeline

3. Issues in your plan

In the **Scope** section, Advanced Roadmaps shows (from left to right):

- issue ranking (if applicable)
- issue type
- issue key
- issue summary

If an issue contains children, select the arrow (**>**) to expand collapsed rows. Or use the **Issue** dropdown menu at the top to expand or collapse all issues.

If you create an issue using the **+ Create issue** button, Advanced Roadmaps places your new issue at the bottom of your current project. If you've made a child-level issue (such as a story or subtask), it will be listed under the collapsible **x issues without parent** heading at the bottom of your timeline until it's linked to a parent task.

4. Issue values in columns

By default, newly created plans include **target start**, **target end**, and **status** columns. You can add columns for values that are already configured in your plan via the **Fields** dropdown menu. Learn how to add, edit, and remove columns in Advanced Roadmaps.

To change the value of these fields, simply select the value you want to change, then press Enter when you're done. Advanced Roadmaps adds an orange accent (displayed in the image above) to the corner of any field which has a change that hasn't been committed to a Jira issue. If you're planning in a scenario, this accent mark will be the color that you chose to represent it.

5. Timeline schedule bars, dependencies, and releases

To the right of the fields columns, you'll find the timeline bars which represent the expected duration of scheduled issues.

The issue bars themselves also hold clues to the structure of your plan. Striped bars indicate that an issue has rolled-up values as discussed above, and the link icons that display on either end of an issue imply that an issue has a dependency attached to it. Learn more about dependencies.

Advanced Roadmaps also displays health indicators on your timeline of your plan including sprint capacity (not pictured; more information can be found on the Monitor capacity from your timeline page) and releases that are included in your plan, which show as red and green circles above your timeline. Learn more about releases on the Monitor Releases section.

6. Clean up your plan and save changes

When you're ready to put your plan into motion, use the **Review changes** button to save the values in your plan back to Jira, overwriting existing values in the process. This will also create issues, teams, and releases that you made from your timeline in your Jira project. Until you do this, every change that you make in Advanced Roadmaps only lives in your plan. Read more about committing your changes back to Jira issues.

If you're planning in a scenario, Advanced Roadmaps saves the values from the scenario you're currently viewing.

Warnings in Advanced Roadmaps alert you to any issues that break your plan. For example, if you have a dependency with misaligned dates. Learn more about Warnings in Advanced Roadmaps in Advanced Roadmaps.

When you're ready to share your plan with the world (or your coworkers) use the **Share as** button to embed your page, create a sharable link, or export as a .csv file. Learn more about the ways you can share your Advanced Roadmaps data on the Share plan and export data page.

Advanced Roadmaps also comes with an **Auto-schedule** function that automatically creates a plan based on issue details and settings. Learn more about the auto-scheduling function of Advanced Roadmaps on the Auto-schedule issues in Advanced Roadmaps page.

> ⓘ Advanced Roadmaps is now part of Jira Software Data Center. To get access to the features described on this page, upgrade to Jira Software Data Center 8.15 or later. Can't upgrade yet? Depending on your current Data Center version, you can access these features by installing the latest version of the app (at no cost). See our FAQ for all the details

View issue details in Advanced Roadmaps

You can choose to display or hide any of the following issue details in your plan:

- Details pulled from issues in Jira Software
    - assignees, sprints, estimates, status, priority, dependencies
- Custom fields in Advanced Roadmaps
    - target start, end dates, teams
- Custom fields configured in Jira Software
    - single-line text fields

You must save any custom fields created in Advanced Roadmaps to Jira Software in order to sort by that field. Learn more about saving changes to Jira Software.

Add, remove, and re-order columns

If you'd like to add more issue details to your plan, use the **Fields** menu to add the relevant fields. These fields will be displayed as columns, containing issue details pulled from Jira Software.

The fields in this menu correspond to the fields currently visible on your plan. Using this menu, you can remove a field, or drag and drop fields to change the order in which they're displayed.

To add a column:

1. Go to **Fields** > **Choose fields…** to select what you'd like to display in your plan. You can also start typing to find a specific field.
2. Select the field you wish to add or press **Enter** to add it to your plan.

To remove a column, select the **x** next to the field in the **Fields** menu.

Resize columns

You can increase or decrease the width of your columns by clicking and dragging the left or right edge of the column. To collapse or expand a single column, click the column header.

Change a value in your plan

The values for the columns in your plan may depend on what's configured in your Jira Software instance. For example, your priority options will depend on the configuration of priority schemes in Jira Software.

While viewing your plan, you can change a value in most columns (excluding status and progress fields) by selecting the cell you want to edit. Then press Enter to save changes to your plan. Note that your changes won't be applied to issues in Jira Software unless you save them.

Filter issues in Advanced Roadmaps

You can search for specific issues on your timeline or narrow the scope of what's displayed in your plan by filtering issues using the **Filters** menu located next to the hierarchy dropdowns. You can also tell Advanced Roadmaps to show full hierarchy and and auto-expand issues while filtering.

Search for a specific issue

You can search for issues in two ways. The first is to use the **magnifying glass icon** at the header of the **Scope** column and enter a word, phrase, or issue key to see all matching issues. Alternatively, a search function can be found in the **Filters** menu labeled **Issue details**.

Regardless of which way you choose to search, you can enter multiple terms separated by a comma with no spaces (for example: IOS-4,IOS-21,TIS,Blocker).

Filters

> ℹ If you're using the Advanced Roadmaps app (version 3.29 or earlier) from the Marketplace, some of the options listed in this section may not be available in your instance.

- **Releases** — See all issues assigned to a specific release.
- **Teams** — Filter by teams in the plan.
- **Assignees** — Filter by assignees in the plan, as well as users which belong to included teams.
- **Sprints** — Show only issues from any sprint included in your plan. You can also filter by type of sprint (completed, future, external, etc).
- **Projects** — Show only issues from any project included in your plan.
- **Issue types** — Only show issues of a certain type. What's available depends on the Jira issue types mapped to Advanced Roadmaps.
- **Components** — Filter by components, assuming they're configured in your plan.
- **Labels** — Filter by labels, assuming they're configured in your plan.
- **Dependencies** — Show only issues with dependencies in your plan. The dropdown menu contains the following options:
- **All issues** — Show all issues included in the plan, regardless of dependencies.
- **Has dependencies** — Show only issues that have dependencies.
- **Specific issue** — Show dependencies related to a specific issue in your plan.
- **Status** — See issues based on their current status and issues created in your plan that are not yet saved in Jira Software.
- **Warnings** — Display only issues that have warnings.

Custom filters

You can filter by custom fields if they're configured for your plan. Advanced Roadmaps supports five custom field types as filters:

- single-choice select
- multiple-choice select
- checkbox
- radio button
- label

Filter settings

You can adjust the filter settings at the bottom of the **Filters** menu to show or hide specific aspects of your plan.

Show full hierarchy

When filtering by issue, search results will only show issues that match your query. The **Show full hierarchy** function will show your results in context of their related issues. Child issues that don't match the filter will appear slightly grayed out.

Show full hierarchy off

Show full hierarchy on

Auto-expand issues while filtering

This setting automatically expands issue hierarchy to reveal issues that match your filter. Without this setting, issues that match your filtering criteria will still be hidden under their parent issues.

Sort issues in Advanced Roadmaps

By default, issues in a plan are sorted by issue ranking. You can also sort issues by the following:

- dates
- estimates
- status
- priority
- assignees and teams

> (i) If you're using the Advanced Roadmaps app (version 3.29 or earlier) from the Marketplace, some of the options listed in this section may not be available in your instance.

Configure sorting order

Navigate to the **View settings** menu, then use the **Sort by** dropdown menu to change how issues are sorted in your plan.

To toggle between ascending and descending order on your roadmap, select the up or down arrow next to the field by which you'd like to sort your plan. If issues are already sorted in ascending or descending order, a reset icon will appear at the top of the **Fields** column. This icon will revert issues to the default rank order.

Rules on sorting issues

Regardless of how sorting is configured in a plan, the following rules still apply:

- The order of issues is determined by the sorting method, and will overrule any drag-and-drop adjustments you make to your plan.
- Issues will be sorted based on the highest hierarchy level shown in your timeline. Child issues will be sorted in the same manner, but still grouped under their parent issue.
- Issues in your plan will be sorted by the field of your choice, regardless of changes you make to other fields.
- Newly created issues in your plan will be sorted according to your defined sorting rule.

Sort issues by:

Dates

There are three dates by which you can sort your issues: target dates, due dates, or custom dates. These different dates are all treated the same way in Advanced Roadmaps, so we simply refer to them as dates.

When sorting issues by dates, keep the following rules in mind:

- Sprint dates will be used if an issue doesn't have a start or end date but is assigned to a sprint.
- Adjusting the dates of issues will change how these issues are sorted in your plan.

Estimates

Regardless of whether you're using story points or time estimates (days and hours), the following rules apply:

- Issues without an estimate will be sorted as though they have an estimate of zero.
- Issues that have rolled-up estimate values will be sorted by the total estimate value of a parent issue plus its children.
- If you change the estimation units of your plan, issues will revert to the default sorting order.

Status

When you sort issues by status, the following rules apply:

- Issues will be sorted by status within their parent hierarchy level. Issues with the same status type will be ordered alphabetically.
- Issues created in your plan that haven't been saved to Jira Software are sorted above those already saved when in ascending order.

Priority

To sort issues by priority, the projects in the plan must use the same priority scheme. If they're not, the option to sort by priority will be disabled.

Assignees and teams

Unassigned issues will be grouped together at the top or bottom of your plan, depending on whether you're using ascending or descending order.

Preconfigured timeline views in Advanced Roadmaps

ⓘ The ability to create multiple views was introduced in Advanced Roadmaps version 3.23. If you're using an earlier version of Advanced Roadmaps, this process may be different or not available in your instance.

Advanced Roadmaps lets you swap between saved view configurations to focus on different aspects of your plan. Each view shows different columns of information on your timeline, though you can modify these views as you see fit. The four preconfigured views are:

- Basic view
- Sprint capacity management view
- Top-level planning view
- Dependency management view

You can cycle through these using the **Views** menu under the **Review changes** button:

In all of the preconfigured views, **Roll-up to parent** is disabled, and all issues will be collapsed and grouped according to the view settings. When you change a setting, an ⟨EDITED⟩ lozenge will appear in the **Views** menu to indicate that this view has been modified but not saved. Learn more about how to modify a view.

Basic view

Basic view is the default view for new plans.

How to read this view:

1. Columns: **Start date, due date,** and **status**
2. Three month timeframe
3. Hierarchy levels set from highest to lowest, per hierarchy configuration
4. Issues colored by status
5. Issues sorted by ranking

Sprint capacity management view

The Sprint capacity management view shows capacity across issues in a plan so you can optimize sprints or iterations.

How to read this view:

1. Columns: **Priority**, **sprint**, **story points (estimates)**, **team (collapsed)**, and **release (collapsed)**
2. Three month timeframe
3. Hierarchy levels set to **Story**

4. Issues filtered by **All future sprints**
5. Issues grouped by teams with capacity showing in the timeline
6. Issues colored by status
7. Issues sorted by ranking

Top-level planning view

The top-level planning view gives you a high-level view of how work is progressing by showing issues from highest to lowest hierarchy level. You can also see issues in their hierarchical relationship with each other.

How to read this view:

1. Columns: **Progress (issue count)**, **start** and **due date**, **release**, and **component**
2. One year timeframe
3. Hierarchy level set from the highest level to **story level**, per hierarchy configuration
4. Issues grouped by projects
5. Issues sorted by ranking
6. Issues colored by teams

Dependency management view

This view focuses only on issues that have dependencies in a plan so you quickly see which ones are blocking the progress of others.

In the screenshot below, some issues don't match the view's filters so they appear grayed out in the timeline.

How to read this view:

1. Columns: **Dependencies** (incoming and outgoing), **priority**, **start** and **due date**, and **team**
2. One year timeframe
3. Hierarchy set from epic to sub-task
4. Filtered by issues with dependencies
5. Issues colored by status
6. Issues sorted by ranking

Modify a view in Advanced Roadmaps

As you use Advanced Roadmaps more, you may want to modify the preconfigured views that are included by adding a custom field. Advanced Roadmaps lets you modify a view, then save it so you can access it when needed.

To begin, use the **Views** dropdown menu to select which view you want to modify.

Then use the **View settings** menu to group issues, show capacity, add color, roll-up values, view warnings, and sort issues on your plan. After you make a change to the view, an ⬜EDITED lozenge will appear after the name to indicate that your modifications haven't yet been saved.

Save your view

You have two options when it comes to saving your view, both of which can be accessed by clicking **more** ••• next to the **Views** dropdown menu.

Selecting **Save** will apply the changes you've made to the view selected, overwriting the previous version.

If you select **Save As**, Advanced Roadmaps will save your settings as a new view and prompt you to name it. You also have the option to display this view to any user accessing your plan for the first time by selecting the **Make default for everyone** checkbox before saving.

Simultaneous edits

If another user is working on the same view as you and makes a change to it before you save, a modal with two options will appear. You can either **Overwrite the changes they've made** which will get rid of their changes in favor of yours. Alternatively, you can **Save your changes as a new view** which will preserve the other user's changes and save yours in a new, separate view.

Once you've made your decision, select **Confirm** to continue.

Untitled [Draft] views

You may also see an **Untitled [Draft]** view while accessing plans in your instance. There are two reasons you'll see this:

1. You're accessing an existing plan with issue filters and view settings that were previously configured but not saved. You can finalize this draft by saving it as a new view.
2. Someone else deletes your view as you're working on it.

Discard changes made in a view

While making changes to a view, you may realize that you no longer need these changes. If you discard your changes, they will be permanently deleted and the view will revert to its last saved configuration.

To discard changes:

1. In the ••• menu next to the Views dropdown, select **Discard changes**.
2. In the confirmation dialog, select **Discard changes**.

Manage views in Advanced Roadmaps

There are two ways to access views in your Advanced Roadmaps plan. The first is to navigate to the **Settings** menu (gear) > **Configure**, then select **Saved views** from the sidebar. Alternatively, you can select **Manage views** from the **Views** dropdown menu.

Either method will bring you to the following screen:

1. **View titles** - Each view will show when it was last updated and by which user.
2. **Current view** - The label (current) shows which view you're currently using.
3. **Delete plan icon** - Exactly what it says on the tin. This will delete the selected view. Note that you can't recover a deleted plan view and you can't delete the current default view.
4. **Current default view** - The view a user will see when they load your plan for the first time.

> (i) Default views were added as part of the 3.20 Advanced Roadmaps release. If you're using an earlier version of Advanced Roadmaps, this feature won't be available to you.

If a view has been edited and hasn't been saved, it will display the EDITED lozenge as detailed on the Modify a view page.

Rename a view

> (i) The ability to rename a view was introduced in Advanced Roadmaps version 3.28. If you're using an earlier version of Advanced Roadmaps, this process may be different or not available in your instance.

Views can be changed as the goals of your plan and organization evolve. Select the **title** of the view you wish to rename, then select the **checkmark** to save your changes.

Delete a view

Deleting a view will permanently remove it from the plan. Anyone currently using the view will no longer be able to use it.

1. After locating the view you want to delete, select the **trash icon** under **Actions**.
2. In the confirmation dialog, select **Delete**.

Monitor progress from your Advanced Roadmaps timeline

Once you've created your plan in Advanced Roadmaps and the work has begun, you can track progress of your projects in a variety of different ways:

Estimates and issues - Track your work based on the number of issues completed, estimates consumed, and more.

Capacity - Balance the workload of a project among your team to ensure you'll meet your deadlines.

Releases - If your project is part of a release, you can find out whether your project is on track to meet your intended release date.

Dependencies - Get deeper insight into which tasks are blocking others and adjust your plans to meet deadlines.

Monitor estimates and issues from your timeline

You can set Advanced Roadmaps to show you the progress of your plan based on different metrics as part of your timeline view. Any changes that haven't been saved to Jira Softwarewon't be taken into account to calculate progress.

You can add the following columns to track your progress:

- Status column - (**TO DO** , **IN PROGRESS** , and **DONE**)
- Progress (estimate)- Shows progress in story points or days depending on plan settings
- Progress (issue count) - Shows the number of child issues remaining to be completed

To learn more about adding columns to your plan, see View issue details in Advanced Roadmaps.

Monitor estimate-based progress

Once you've added the **Progress** (**story points** or **days**) column to your plan, hover on the progress bar to show a flyout window:

This breakdown of work takes into account the estimates of all child issues as well as the parent. You'll also see a breakdown of estimated and unestimated issues. Any unestimated issues may impact the accuracy of your progress. If you've enabled rolled-up values in a plan, you may see inconsistent values between rolled-up estimates of a parent issue and the values in the progress bar.

Monitor issue count progress

Once you've added the **Progress (issue count)** column to your plan, hover on the progress bar to show a flyout window:

This breakdown of work displays the number of issues in each status category and the total number of issues in the plan. The estimates for each issue are not included since this column only covers issue count.

Monitor capacity from your timeline

> ⓘ Capacity and velocity planning was introduced in Advanced Roadmaps version 3.28. If you're using an earlier version of Advanced Roadmaps, this process may be different or not available in your instance.

Before you can monitor capacity from your timeline, you must enable it in your plan. Learn how to do this on the Capacity and Velocity in Advanced Roadmaps page.

Read capacity on your timeline

Once you've added capacity to your timeline, you'll see a sprint status bar with extra tabs that each correspond to a sprint. The green bar in the sprint tab indicates current capacity. If you continue to assign issues to that sprint, the bar will turn red when the work assigned to the sprint exceeds its capacity. Advanced Roadmaps only shows capacity for current or future sprints. Learn how to manage capacity.

Select a sprint on the timeline to bring up the sprint capacity window:

1. Sprint information - Name, start and end dates, status lozenge, sprint goal (as defined by administrator), and total sprint capacity
2. Progress bar - Story points allotted and completed
3. Further viewing options for this sprint

The number of unestimated issues in your sprint will appear in the sprint capacity window next to the label **Unest imated issues.** Issues that have due dates within your sprint but are not assigned to it will appear next to the **x unassigned issues during sprint** label.

Sprint states

Sprints are represented in five different states: past sprints, completed sprints, active sprints, future sprints, and projected sprints.

Past sprints

Past sprints are unnamed representations of what past iterations would have been if the sprints existed at that point. They're created based on sprint length and are used for demonstration purposes only. The sprint capacity window won't appear when you hover over the sprint status bar.

Completed sprints

While past sprints are representations of past iterations, **Completed sprints** are scheduled sprints that have been completed by your team. You can hover over completed sprints to see details, but you can't change the information.

Active sprints

An **Active sprint** is the sprint that's currently in progress and has an ACTIVE SPRINT lozenge next to the dates.

When auto-scheduling work in a plan, issues will be scheduled into the next available future sprint, based on the remaining sprint capacity. Issues won't be assigned to a currently active sprint.

If there's any unexpected change in a team's capacity (for example, a team member is sick), you can quickly update the sprint capacity in the active sprint's menu.

Future sprints

> (i) Future sprints were introduced in Advanced Roadmaps version 3.22. If you're using an earlier version of Advanced Roadmaps, this process may be different or not available in your instance.

Any sprint that already exists in Jira Software and is scheduled after the active sprint is called a **Future sprint** and will have a FUTURE SPRINT lozenge next to its name. Even though a sprint can only be given dates when it becomes active, Advanced Roadmaps is able to infer future sprints based on the iteration length.

Projected sprints

Any sprints that are projected to take place beyond the scope of existing future sprints will also be displayed in the timeline. These are called **Projected sprints** and will have a PROJECTED SPRINT lozenge next to the dates.

Monitor releases from your timeline

Advanced Roadmaps gives you several ways to track the progress of releases in your plan.

The Releases tab

The first way to monitor the health of your releases is to navigate to the **Releases** tab. Next to each release, you'll see the status, progress bar (issue count), start and release dates, and any cross-project releases it's included in, as demonstrated below:

From this view, you can get further detail about each release by viewing the release on your roadmap or viewing it in Jira Software. You can also hover on the Progress bar to bring up a window that breaks down issues in that release according to status.

Monitor releases from your timeline

You can also track the progress of each release from your timeline. The status of releases are indicated by icons that correlate to the release date. A green icon means that a release is on schedule while red icons indicate that a release is off-track. To get your release back on track, adjust the dates of issues or reschedule the release. Plans that contain issues that schedule work by sprint dates will be shown as off-track if the sprint ends after the release is scheduled to be shipped.

Hovering on an icon will bring up a window that gives you a high-level overview of your release. Use the left or right arrow at the top of the window to quickly jump between releases, as seen below:

Sometimes, the icon will have a number above it. This is to tell you that there are multiple releases happening on that day. Use the same left and right arrows in the window to cycle through them.

Lastly, you can highlight issues contained in a release on your timeline. Select the **Highlight on timeline** check box, and the release will be displayed with either a red or green line in the timeline.

Filter issues by release

Advanced Roadmaps comes with several powerful view settings that let you **Filter issues by release** and **by projects**, which is covered in our Filtering issues section.

Monitor dependencies from your timeline

To view the details of the dependencies in your plan, select the badge on either side of the schedule bar as discussed on the Dependencies in Advanced Roadmaps page. This will bring up the dependency details window, which includes:

- **Type** - "is blocked by" denotes an incoming dependency while "blocks" denotes outgoing
- **Issue** - Issue details of the dependent issue
- **Status**
- **Assignee**
- **Lead time** - The number of days between the dependent issues
- **Filter by dependencies** - The issues that comprise the dependent chain will be displayed in your timeline. Learn more about this on our Filter issues in Advanced Roadmaps page.

When viewing dependencies on your timeline, the badge on either end will turn red if there are conflicting dates. To resolve this, change the start or end date of either dependency.

Dependency management view

Advanced Roadmaps comes with a Dependency management view which highlights dependencies on your timeline.

The Dependencies report in Advanced Roadmaps

> ℹ The dependencies report was introduced in the Jira Data Center and Server 8.19 release. If you're using the standalone Advanced Roadmaps app from the Marketplace (version 3.29 or earlier), please refer to the 8.15 version of this page.

The Dependencies Report displays issues from your plan as tiles with arrows to illustrate which issues are dependent on others. To open this view, select the **Dependencies report** tab underneath your plan name.

> ℹ The Dependencies report is a view-only tool. Switching to this view will not change your plan.

The solid gray arrow between issues indicates their relationship and shows that these dependencies have no warnings or errors.

If the dependency generates a warning, such as misaligned dates, the arrow will turn red:

To turn the arrow back to gray, return to your roadmap and resolve the corresponding error. You can easily locate the issue in your roadmap by navigating to the ••• on the issue tile then selecting **Locate issue in roadmap**.

View your dependencies

The Dependencies report gives you three different ways to sort and view your dependencies.

Roll-up to

This feature allows you to choose at which hierarchy level you would like to view your dependencies. In the image below, the story-level task IOS-13 belongs to the epic IOS-6 and is blocking ADR-14, another story-level task, which belongs to the epic ADR-5. Here we see that relationship without roll-up settings applied:

When rolled up to the epic level, Advanced Roadmaps will show a dotted line to indicate that the dependency being referenced is not of the same hierarchy level as your current view. This is because while they have stories that are dependent, the epics are not dependent on each other. The dotted line will turn red if your dependency structure has warnings which require your attention.

When rolled up to the epic level, you can still view which issues are linked with dependencies by selecting the **x links** text above the dotted arrow to show the specific issues represented.

Group by

You can also view dependencies based on the team to which they're assigned by selecting **Team** from the **Grou p by** dropdown. This will group your dependencies based on their assigned team.

In the example above, IOS-13 and IOS-19 are assigned to the IOS board team while ADR-14 is unassigned.

Filter by issue

Lastly, you can **Filter by issue** to highlight a single issue and see a visual layout of its dependencies. In the example below, IOS-13 blocks both IOS-19 and ADR-14:

The **Filter by issue** function lets you focus on one specific issue and dependencies related to it, and can be accessed in two ways. The first is by using the **Filter by issue** dropdown at the top of the map. Alternatively, you can select **Filter by this issue** from the ••• menu.

The example below shows what happens when the function is applied to only show the issues related to IOS-19:

Group issues on your Advanced Roadmaps timeline

Advanced Roadmaps lets you group issues on your timeline according to several different factors:

- assignees
- projects
- component
- teams
- sprints
- release
- labels

Group by assignees

Each assignee will be allocated a swimlane containing assigned issues. Any unassigned issues will be in the Unassigned swimlane.

To group issues by assignees, go to **View settings** on your plan and select **Assignee** from the **Group by** menu.

Group by projects

Each project will be allocated a swimlane containing its assigned issues.

To group issues by projects, go to **View settings** on your plan and select **Project** from the **Group by** menu.

Group by component

> ⓘ Before you can group by components, they need to be configured in your Jira Software project.

To group issues by components, go to **View settings** on your plan and select **Component** from the **Group by** menu. Select **+ Create group**, enter a name, and then start typing to find the components to add to the group.

To delete a component group, go to **Group by component** in **View settings**, then select the **delete** icon next to the group you want to delete.

Group by teams

When you group issues by teams, issues will display in team-specific swimlanes. If you've associated the teams in your plan to issue sources, you can also show the overall capacity of these teams in the timeline. This can help you visualize how team-specific work is distributed across multiple sprints. See Monitor capacity from your timeline for more information.

To group issues by teams, go to **View settings** on your plan and select **Team** from the **Group by** menu. Select the **Show capacity on timeline** checkbox to display the sprints associated to teams in the plan.

Group by sprints

When you group issues by sprints, the issues will be grouped and displayed into sprint-specific swimlanes. You can expand these swimlanes to view the issues for each sprint. If a sprint is shared by multiple teams, each team gets its own sprint swimlane and the capacity shown will only be for that specific team.

Grouping by sprint will automatically Roll-up values to the parent issue. You can also choose to show the capacity for each sprint in the timeline.

To group issues by teams, go to **View settings** on your plan and select **Sprint** from the **Group by** menu. Select the **Show capacity on timeline** checkbox to display the sprints associated to teams in the plan.

Group by release

When you group issues by releases, issues will display in release-specific swimlanes. You'll see the release icon beside the release name followed by the project name.

Releases are ordered by release date followed by those without a date. If an issue isn't assigned to a release, it's grouped as **Unassigned** at the bottom of the plan.

To group issues by teams, go to **View settings** on your plan and select **Release** from the **Group by** menu. You can change release assignment by using the dropdown in the **Release** field on your timeline.

Group by labels

> ⓘ Before you begin, the labels must be configured by your administrator. Learn how to configure custom fields on the Configure your Jira Software instance for Advanced Roadmaps page.

To group issues by labels, go to **View settings** on your plan and select **Label** from the **Group by** menu. Select **+ Create group**, enter a name, and then start typing to find the labels to add to the group.

To delete a label group, go to **Group by label** in **View settings**, then select the **delete** icon next to the group you want to delete.

Add colors to issues on your Advanced Roadmaps timeline

In the Roadmap view of your plan, issues are displayed on the timeline as schedule bars. The size of each bar corresponds to the issue's start and end dates. By default, schedule bars are gray but you can apply different colors to them for visual differentiation or to surface trends.

Options for coloring

You can add color to schedule bars based on the following:

- Project
- Status
- Team
- Components — Only available as an option if the components have been configured.
- Labels — Only available as an option if the labels have been configured.
- Custom fields — Only available if custom fields are enabled and have been configured.
- Issue type — Based on the issue types configured for the projects being used as issue sources.

> ⓘ If you're using the Advanced Roadmaps app (version 3.29 or earlier) from the Marketplace, some of the options listed in this section may not be available in your instance.

With these last four options, you can choose the colors, and it's also possible to assign one color to multiple options. The color for project, status, and teams can't be changed.

Add color to your timeline

To color schedule bars by one of the options listed above:

1. In the Roadmap tab, select **View settings**.
2. Use the **Color by** menu to select an option.
3. For options that allow you to pick a color, select **+ Add color** to choose one.
4. Use the dropdown menu to select one or more options for that color.
5. When you're done, select **View settings** again to close the menu.

Return to this menu when you want to update or change the color-by settings applied to your plan. There's also an option to **Clear all colors**.

To switch back to the default gray, use the **Color by** menu to select **None**.

Share plan and export data from Advanced Roadmaps

When you're ready, you can export your data to a .csv file that you can import into a different program such as Excel, or share your plan with relevant stakeholders in a variety of ways.

Share your plan

When you share your plan, any view settings that you've applied will be visible to all viewers. For this reason, we recommend cleaning up your view, including turning off warnings, before sharing your plan.

There are three different ways to share a plan:

- Embed your plan using the Confluence macro
- Create an embed code for a non-Confluence page
- Share the direct link
- Export your data to a .csv file

Embed your plan on a Confluence page

> ℹ️ The Advanced Roadmaps in Confluence app is included with Confluence DC versions 7.20 and later.
>
> If you're using a Confluence DC version between 6.5.0 and 7.19, your Confluence administrator must download and install the app from the Atlassian Marketplace. Ensure that your Confluence DC instance is using version 1.2.3 or newer. View and download in the Atlassian Marketplace.

The Advanced Roadmaps plans macro for Confluence automatically updates your Advanced Roadmaps plan whenever you save your changes.

> 🚫 It's possible to use the HTML macro to add your roadmap to a Confluence page. However, we do not recommend using this method as it makes your Confluence instance vulnerable to cross-scripting attacks. Learn more.

To use the macro in a Confluence page:

1. On the Roadmap tab, select **Share** (next to **Review changes**).
2. In the left menu, select **Confluence** > **Generate link**, and copy the generated link.
3. Navigate to the page where you'd like to insert your roadmap. Insert a macro, select the Advanced Roadmaps **plan** macro, and paste the generated link from the previous step.
4. Choose a height setting for the macro to display on the page, then select **Insert**.

> ℹ️ We recommend a minimum height of 700px for the best viewing experience.

Embed your plan on a non-Confluence page

You can embed your current plan in a non-Confluence page by generating embedded element code. To generate the code:

1. On the Roadmap tab, select **Share** (next to **Review changes**).
2. In the left menu, select **Embed**. This will generate an embed code that you can add to any page, though viewers must have permission to view your plan.

Share direct link

To share a direct link for your plan:

1. On the Roadmap tab, select **Share** (next to **Review changes**).

2. In the left menu, select **Link**. This will generate a link that you can share with anyone, though they must have permission to view your plan.

Export your data

> ⓘ The Export your data to a .csv file function was introduced in Advanced Roadmaps version 3.23. If you're using an earlier version of Advanced Roadmaps, this process may be different or not available in your instance.

To export your plan data, go to **Export** > **Spreadsheet (.csv)**. This will generate a .csv file of the plan's data based on the current filters and hierarchy range applied to the plan.

Administering Jira Software

All administrative functions of Jira Software require you to be a user with the **Jira Administrators** or **Jira System Administrators** global permission.

As Jira Software is based on the Jira platform, many of the functions are documented in the Jira Admin documentation.

Installing Jira Software and upgrading Jira Software	If you want to install a Jira application, or add a Jira application to your existing Jira installation, see Installing Jira applications *(Jira Admin documentation).* If you want to upgrade your current version of Jira, see Upgrading Jira applications *(Jira Admin documentation).*
Backing up Jira	We recommend that you back up your Jira instance regularly. You should also back up before undertaking any activities that could result in unexpected data loss/changes, such as upgrading Jira to a new version or splitting your instance across multiple servers. Learn more: Backing up data *(Jira Admin documentation)*
Supported platforms	We thoroughly test and provide support for our supported platforms (browsers, databases, Java platforms, etc). We strongly recommend that you use the supported platforms for your version of Jira. Learn more: JIRA supported platforms *(Jira Admin documentation)*
Jira system administration	Jira server administration covers server administration (e.g. search indexing, audit logs, etc), global settings (e.g. time tracking, managing shared filters and dashboards, etc), and server optimization (e.g. performance testing, security, etc). Learn more: Jira system administration *(Jira Admin documentation)*
Permissions overview	Jira Software has different types of permissions that cover a range of functionalities. This gives you the flexibility to restrict parts of the application to certain users or groups of users. For example, you could allow one user to assign issues in a project, but prevent another user from doing the same. Learn more: Permissions overview
Configuring projects	This page provides information on how you configure projects in Jira Software, including defining projects, managing versions and components, configuring issues and project permissions, managing project notifications, project schemes, and screens, using the issue collector, working with workflows, etc. Learn more: Configuring projects *(Jira Admin documentation)*
Layout and design	You can configure the layout and design of Jira Software to suit your organization's needs and preferences. For example, you can change Jira's color scheme and logo, set a default dashboard, show an announcement banner, and more. Learn more: Layout and design *(Jira Admin documentation)*

Managing project role memberships

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

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

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

Viewing and editing project role members

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

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

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

Board permissions can be divided into two parts — board administration permissions and board usage permissions.

Board administration permissions cover functionality for changing the configuration of a board. For example, changing columns, customizing cards, etc. Board administration can be assigned to groups or users.

Learn more: Configuring a board

Board usage permissions cover functionality for the usage of a board. For example, creating sprints, ranking issues, etc. Board usage permissions are derived from project permissions. This is described in more detail below.

522

Jira Software functions by permission

In the Jira Software documentation, most configuration options are described as being restricted to either Jira administrators, project administrators, or board administrators.

- A Jira administrator is a user with the **Jira administrators** global permission.
- A project administrator is a user with the **Administer projects** project permission for a particular project.
  By default, the 'Administer projects' permission is assigned to the 'administrators' group (via the Administrators role) for projects.
  Additionally, to perform sprint-related actions, *users need the* 'Manage sprints' permission *for all projects in the origin board — the origin board being the board in which the sprint was originally created.*
- A board administrator is a user that has been added to the **Administrators** for a particular board.
  By default, the administrator of a board includes the person who created it.

Board administration

Function / Functional area	Jira admin	Project admin	Board admin	Notes
Create board	✅	✅	✅	If you create a board via **Boards** (in header) > **Manage Board**, you will not be able to share it, unless you have the 'Create Shared Objects' global permission. If you create the board via the methods below, you do not need the 'Create Shared Objects' global permission to share the board: • Creating a project (where a board is created for the project by default) • Setting up Jira Software for the first time (where you're prompted to create a project, which also creates a board for the project) • Copying a board (the copied board will be shared with the same users as the original board)
Simplify workflow	✅		✅	The board must meet other criteria as well (see Simplified Workflow)
Add status	✅	⚠️	✅	Project must be currently using an Agile Simplified Workflow. You need to be a Jira admin or board admin (to view the board configuration), *In addition*, you need to be a project admin for the one project that is on the board.
Remove status	✅	⚠️	✅	Project must be currently using an Agile Simplified Workflow. You need to be a Jira admin or board admin (to view the board configuration), *In addition*, you need to be a project admin for the one project that is on the board.
All other board configuration functions	✅		✅	

Board usage

Board usage permissions are derived from project permissions.

Depending on the complexity of your board's filter query, you may need further consideration when configuring the 'Manage Sprints' permission for users. For more information on the impact of complex filters, and ways to simplify your filter query, see Using Manage Sprints permission for advanced cases.

Function	Permission Level	Notes
Backlog — Sprints		
Move sprint footer	Manage Sprints permission (for all projects in the board) Schedule Issues permission and Edit Issues permission	
Move issue (reorder /rank)	Schedule Issues permission and Edit Issues permission	Not required if you only move issues across the sprint footer without changing the order of the issues
Start sprint	Manage Sprints permission (for all projects in the board)	Similar permission to creating a Version. Board ownership does not play a role here.
Create sprint	Manage Sprints permission (for all projects in the board)	This permission applies even if the sprint (that is to be started) does not include issues from all projects queried by the board.
Edit sprint information	Manage Sprints permission (for all projects in the board)	Can only be done in the backlog
Reorder sprint	Manage Sprints permission (for all projects in the board)	
Delete sprint	Manage Sprints permission (for all projects in the board)	
Add issue to sprint	Schedule Issues permission and Edit Issues permission	
Active sprints — Sprints		
Add issue to sprint	Schedule Issues permission and Edit Issues permission	
Complete sprint	Manage Sprints permission (for all projects in the board)	Can only be done in Active sprints
Remove issue from sprint	Schedule Issues permission and Edit Issues permission	
Backlog — Epics		
Create epic	Create Issues permission	

Rename epic	Edit Issues permission	
Rank epic	Schedule Issues permission	
Add issue to epic	Edit Issues permission	
Remove issue from epic	Edit Issues permission	
Backlog — Versions		
Create version	Project Admin permission, or Jira Admin permission	
Edit version	Project Admin permission, or Jira Admin permission	
Add issue to version	Edit Issues permission	
Remove issue from version	Edit Issues permission	

Jira Data Center and Server mobile app

A native app for Jira Software Data Center and Server is here!

Manage projects from your fingertips — Find your work, fast — Create and assign issues on the go — Get your work done, wherever you are

Using other Jira applications?

If you're using Jira Core or Jira Service Management, see:

- Jira Core mobile app
- Jira Service Management 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 and boards, 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.
- **Keep the work moving**
  Track the ongoing work on your boards, and transition issues through statuses to get the work done.
- **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** (for Jira 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 page in Play store (only for Android). Simply click the **Open in app** button that displays in mobile web.

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

What you'll need

Jira Software requirements

In order for your users to try the Jira Data Center and Server app, you will need to upgrade your Jira instance to Jira Software 8.3 or 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 15 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 your Jira instance Jira Software 8.3 or later.

Downloading the app

You can download the app for Android (both beta and public version are available) or iOS.

Considerations for administrators

Here are some things to consider when determining whether your users will be able to use the app.

VPN and firewalls

If your Jira instance is not accessible on the public internet, users will need to connect their device to your network or virtual private network (VPN) in order to use the app.

We recommend providing your users with step-by-step instructions on how to connect to your VPN when you let them know the mobile app is available, as this is something Atlassian Support will not be able to help them with.

SSL

The app accepts both HTTP and HTTPS connections.

If your Jira instance is configured to use SSL, you will not be able to log in on the app if:

- your certificate is self-signed and hasn't been installed on the device

- the Certificate Authority (CA) is unknown, or is not one that Android / iOS trusts by default (for example it might be a new CA that is not yet trusted, or a private CA)
- your certificate is missing an intermediate CA, affecting the certificate chain
- your certificate doesn't meet Apple's Requirements for trusted certificates in iOS 13 (affects people using the app on iOS devices).

See our Knowledge base article for information on how to resolve this.

Storage and encryption

The iOS and Android apps cache some content (issues, projects, boards) locally on the device. This helps keep the app responsive when navigating around projects and issues. We don't use any application-level encryption when storing cached data, but the device's internal storage may be encrypted by the operating system.

When a user logs out, all cached data is deleted.

We don't store passwords in the app. Instead we use session cookies, which are encrypted by default.
Login and authentication

The app supports all common Jira user management configurations, including external user directories and SAML single sign-on (NTLM method isn't supported, only basic). Users will need to sign in to use the app, even if your site allows anonymous access.
The app uses an RFC-compliant user-agent header for making the requests during login. This allows us to effectively identify the app and apply further processing based on the user's hardware and software.

Given the wide variety of supported configurations we decided to adopt the most versatile format. For both iOS and Android we compose the User-Agent header value by concatenating the standard WebView User-Agent and unique app identifier.

iOS User-Agent

Format

```
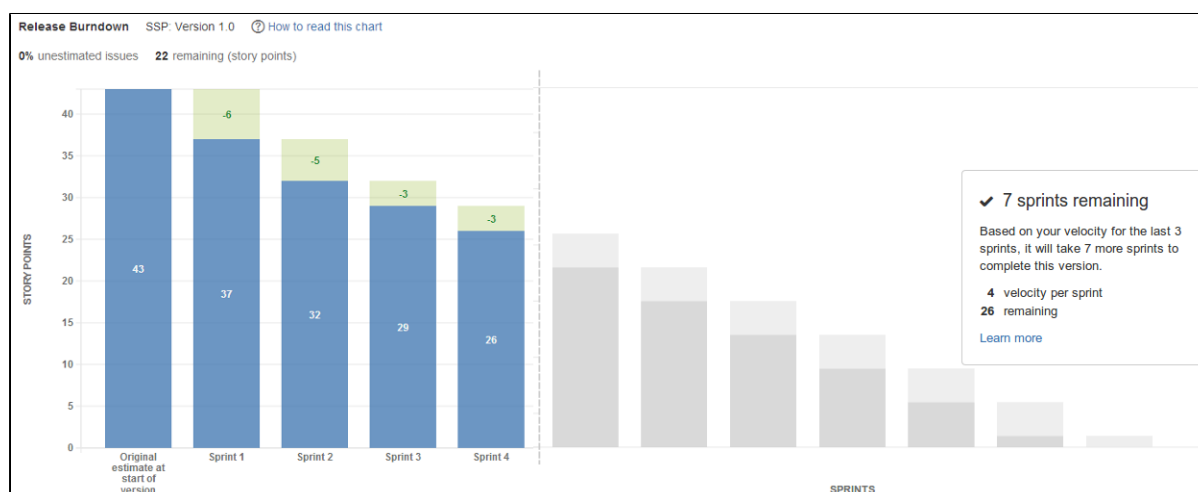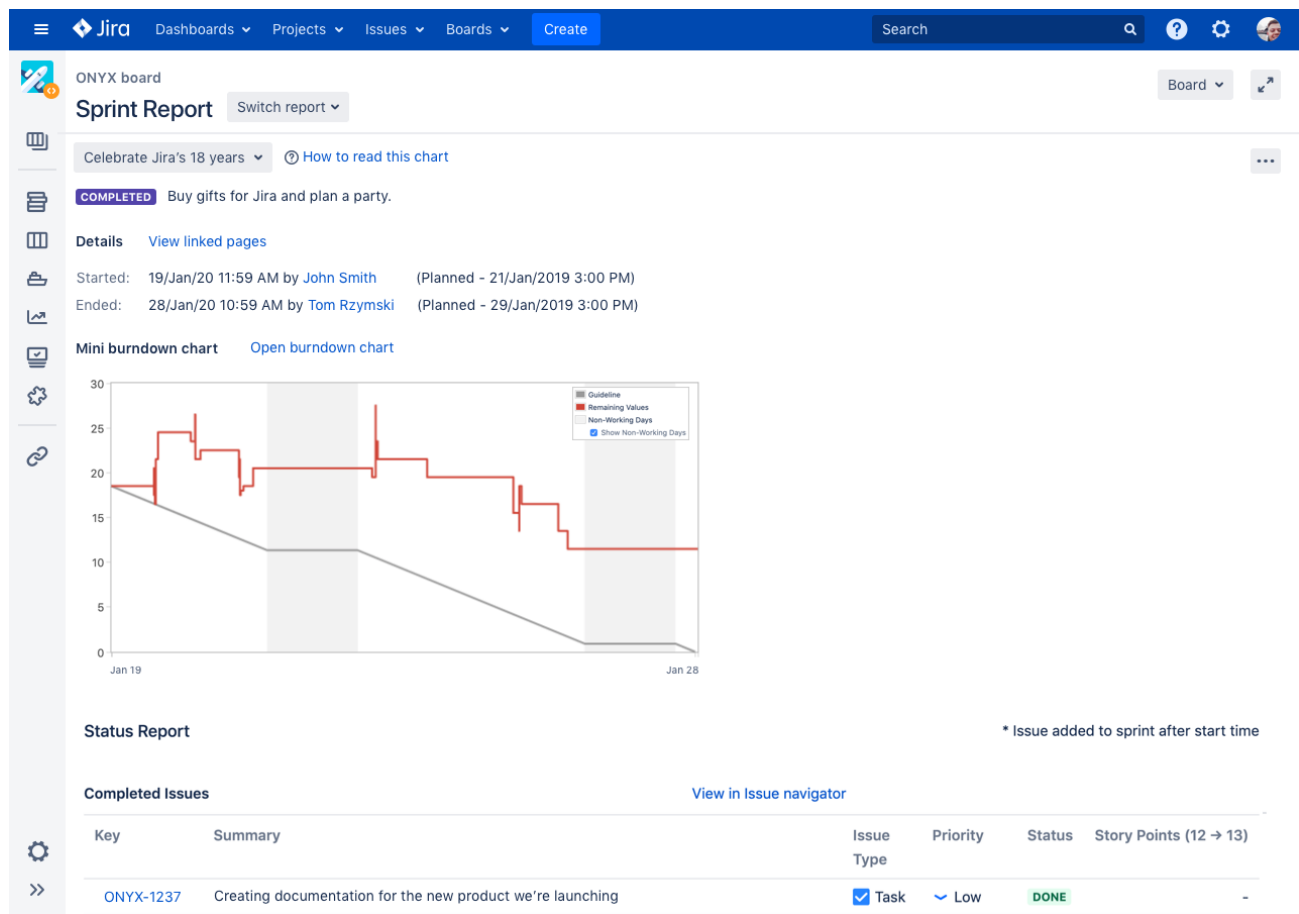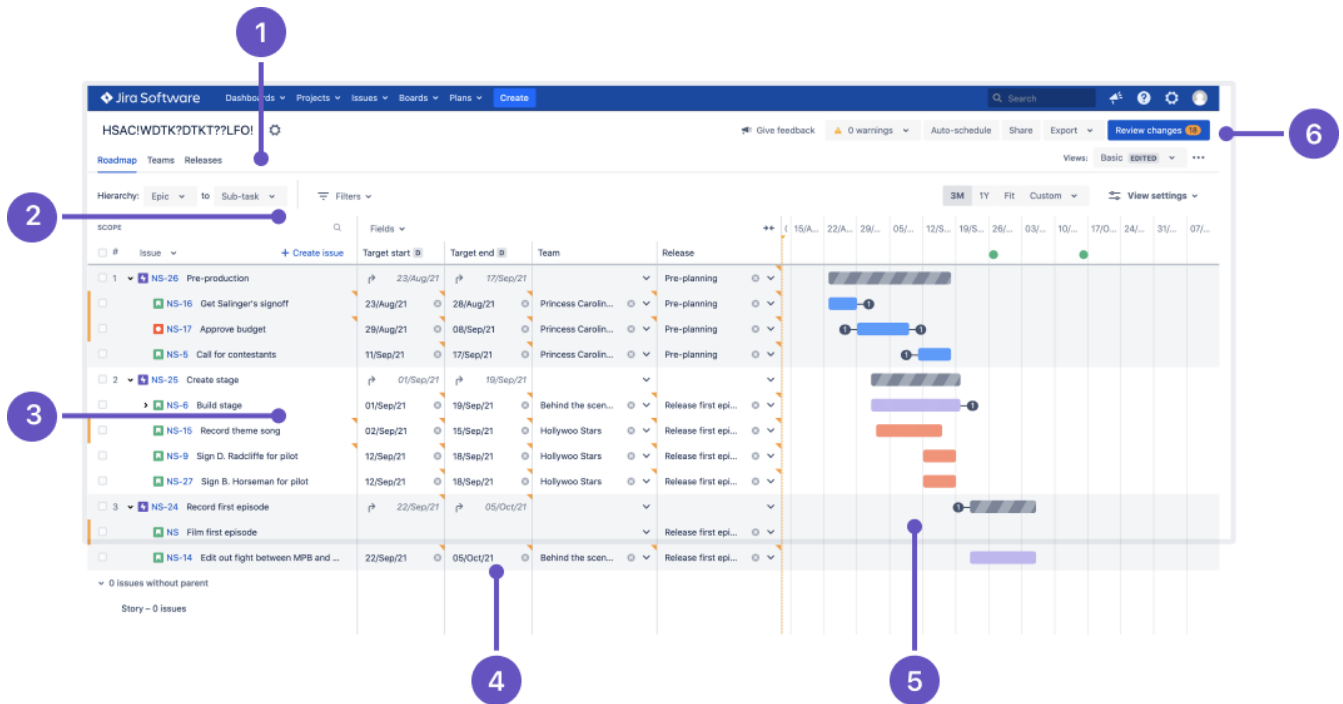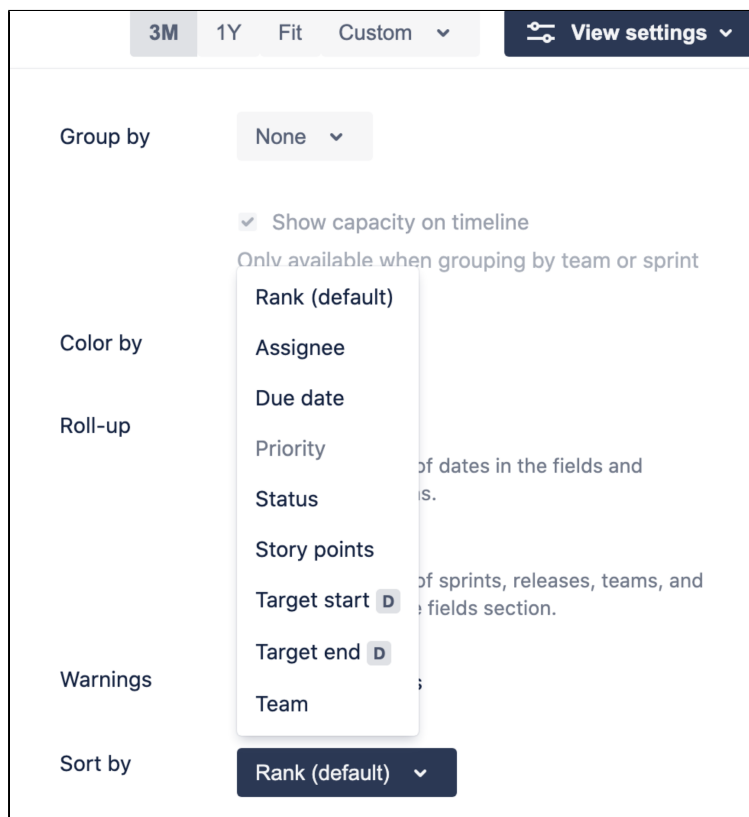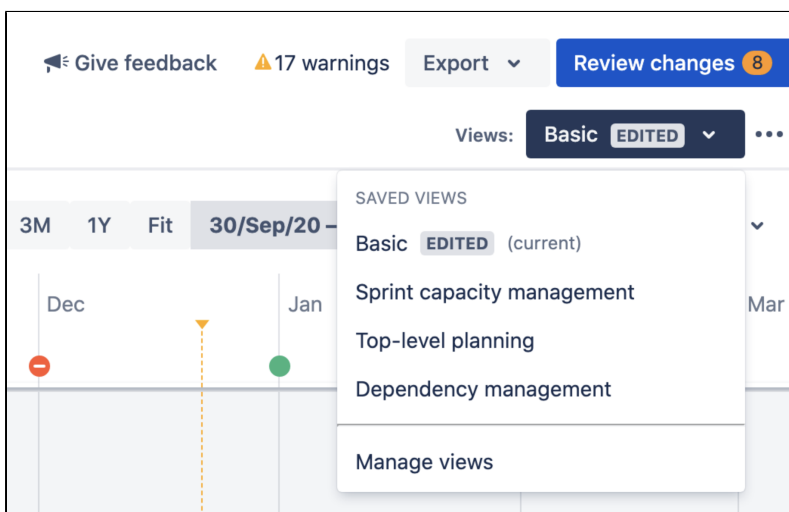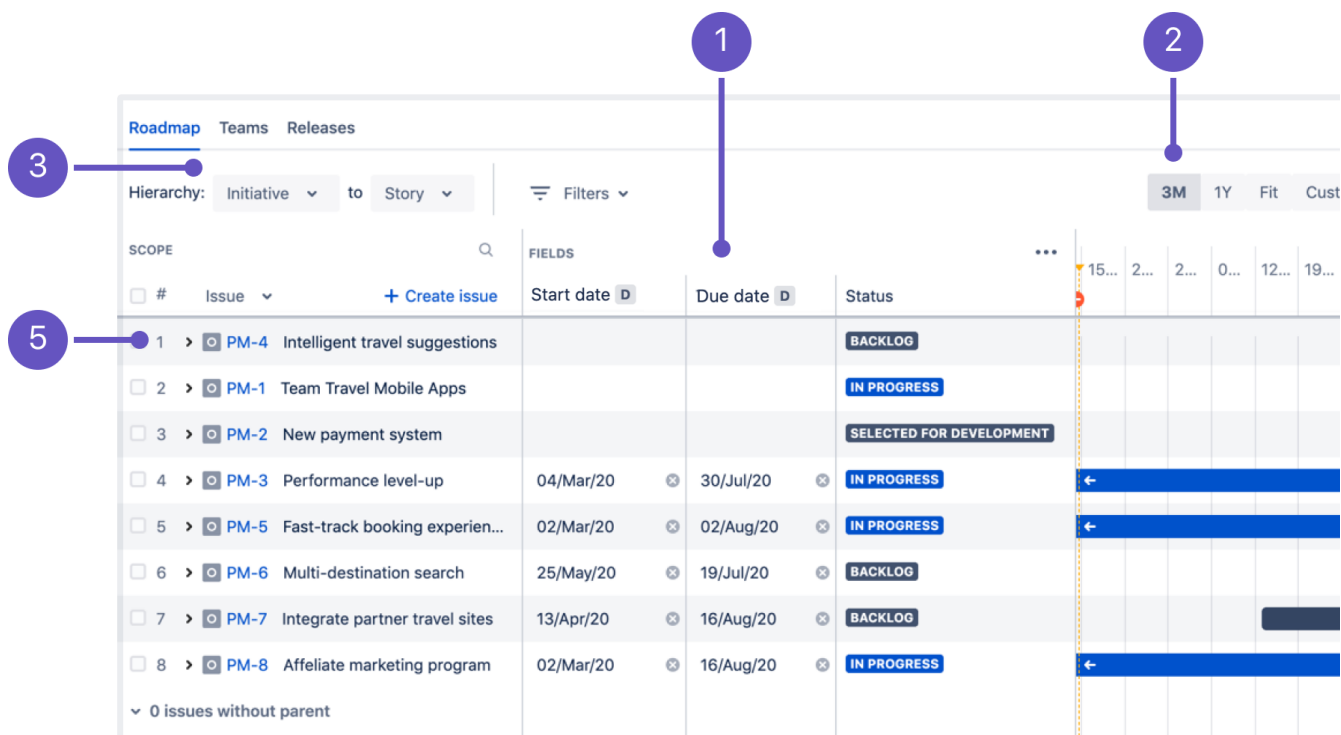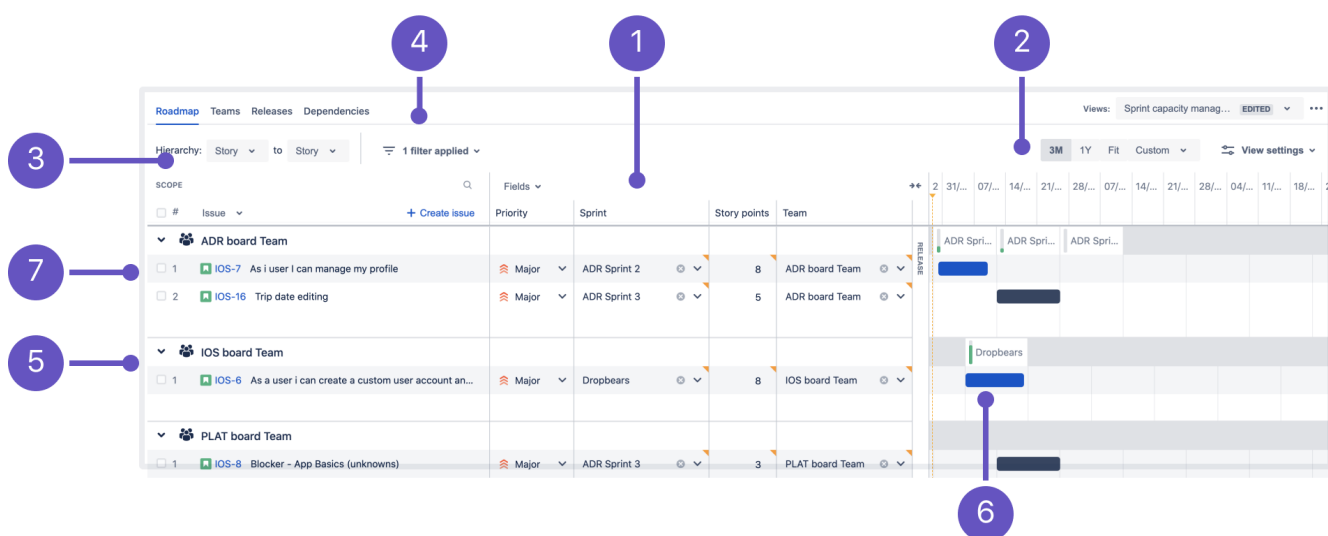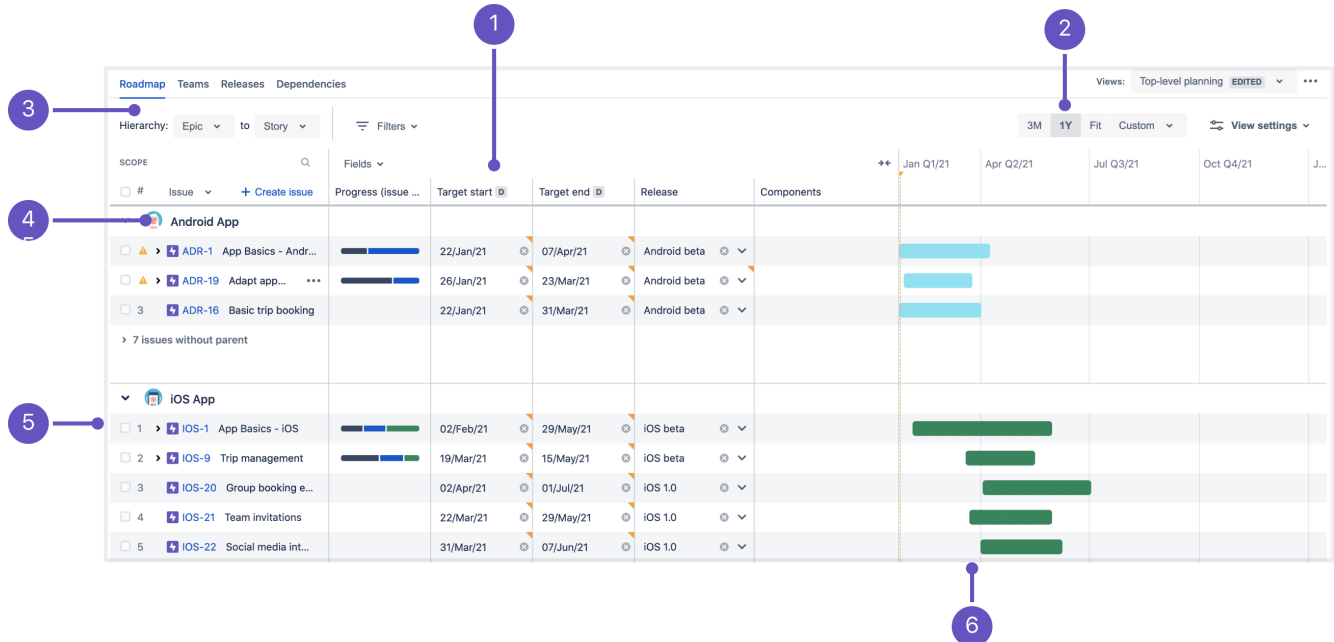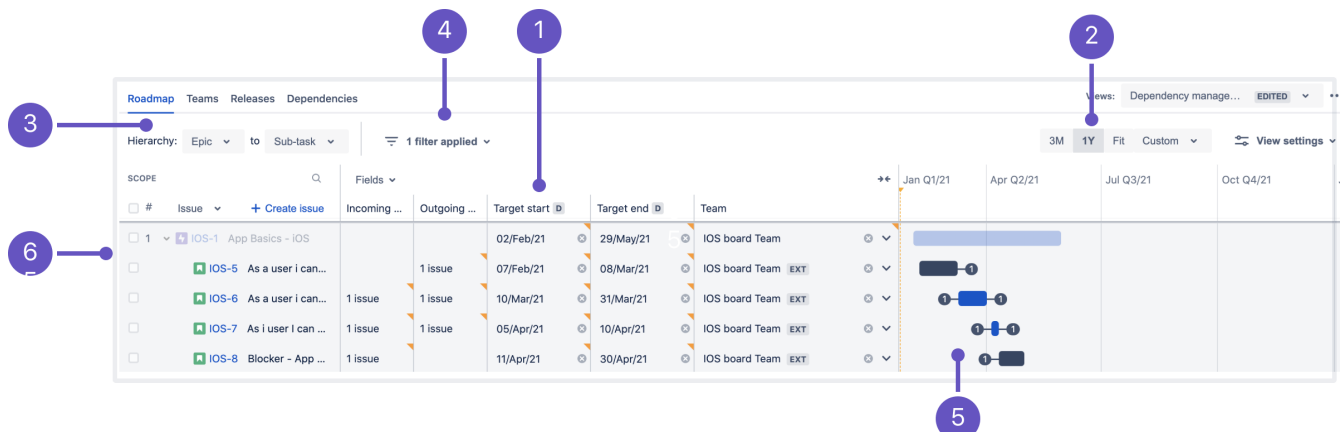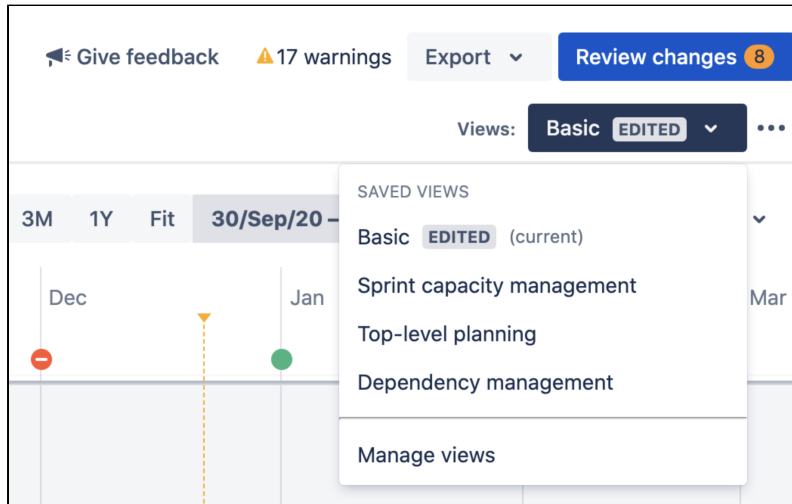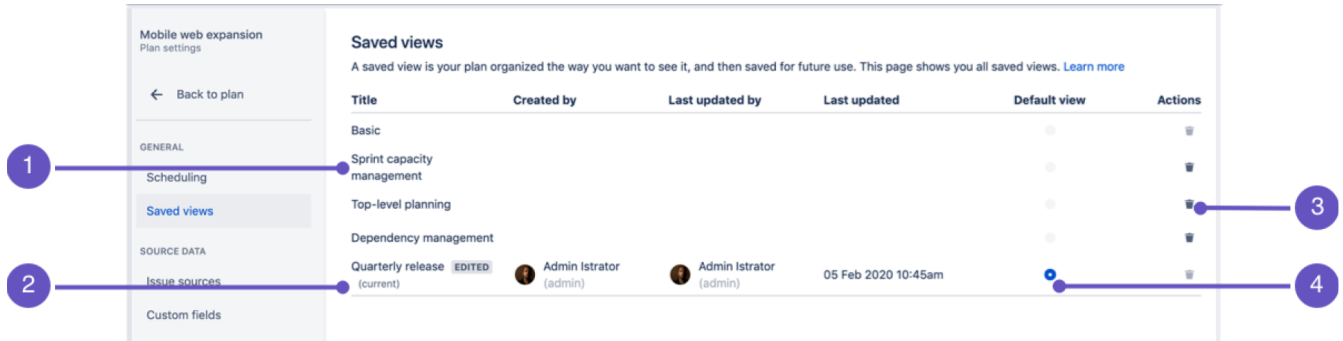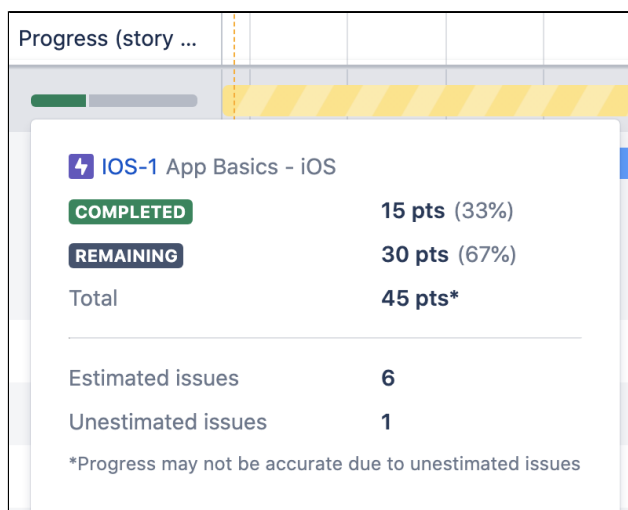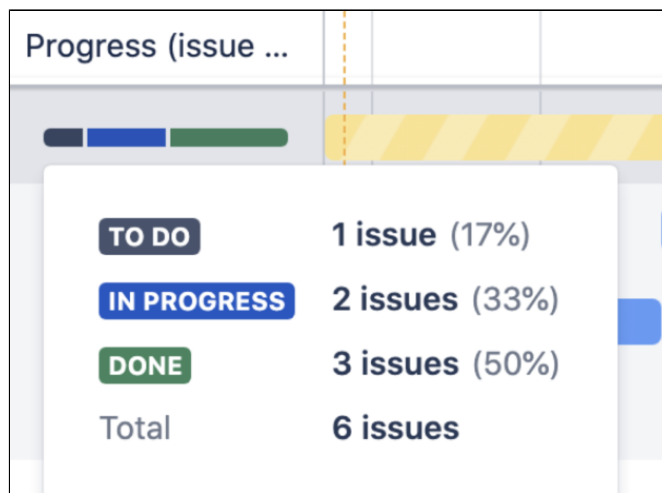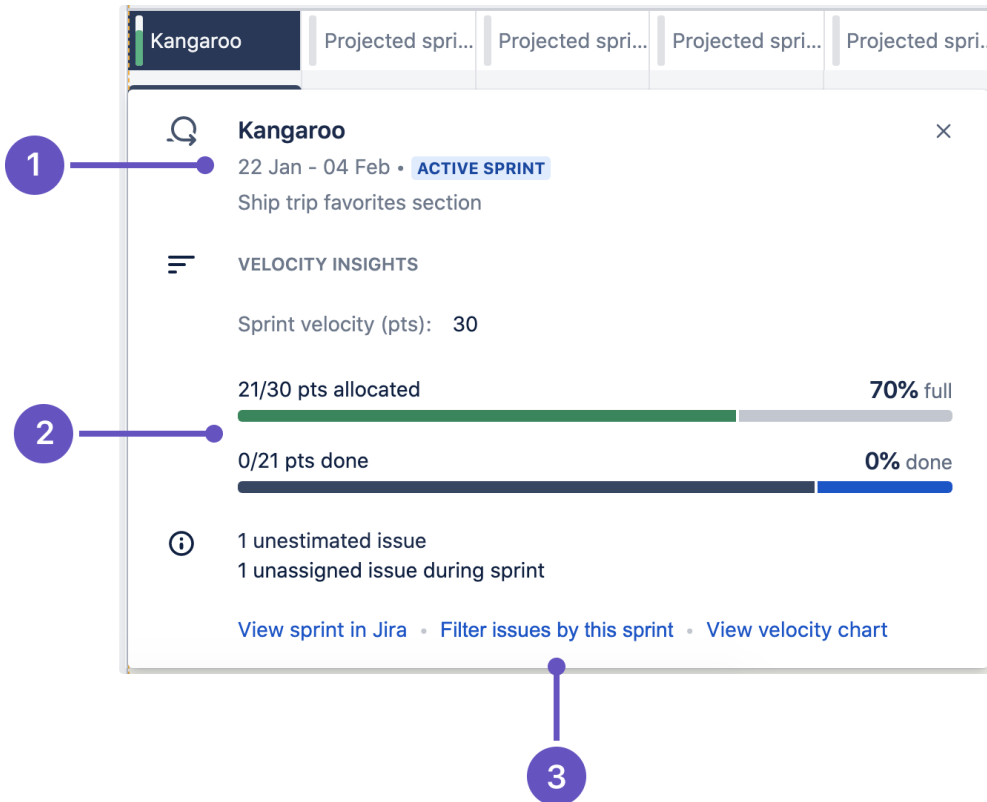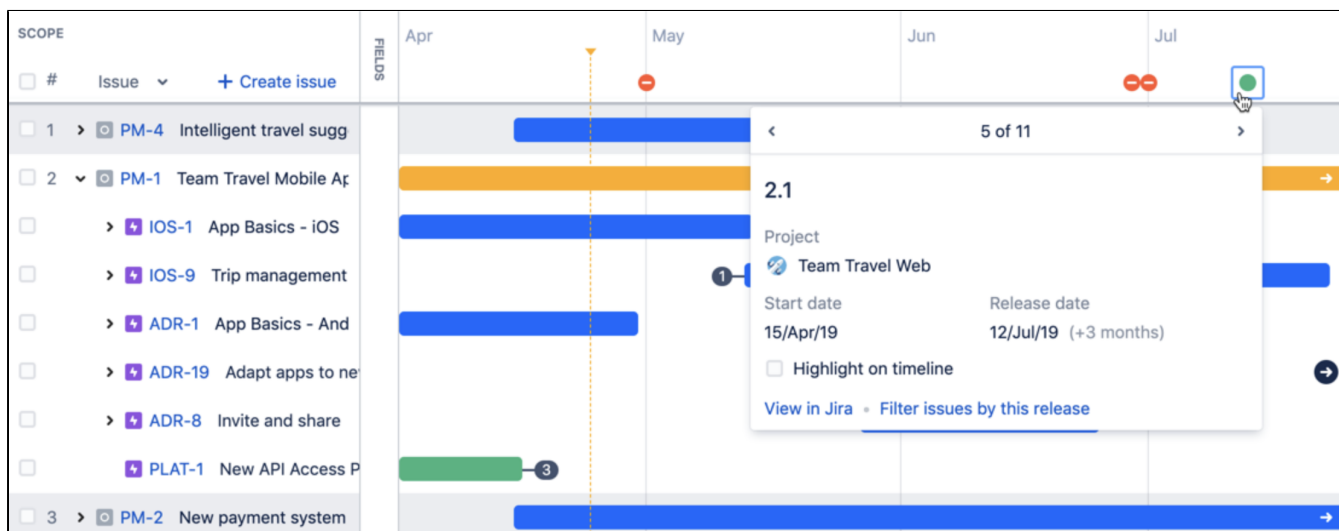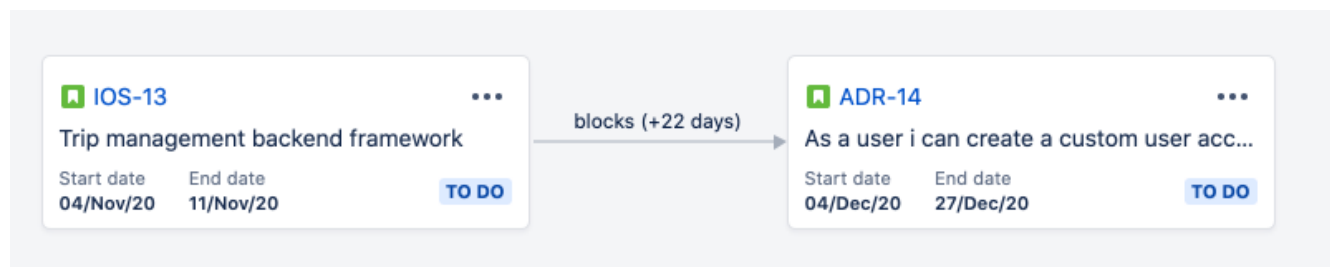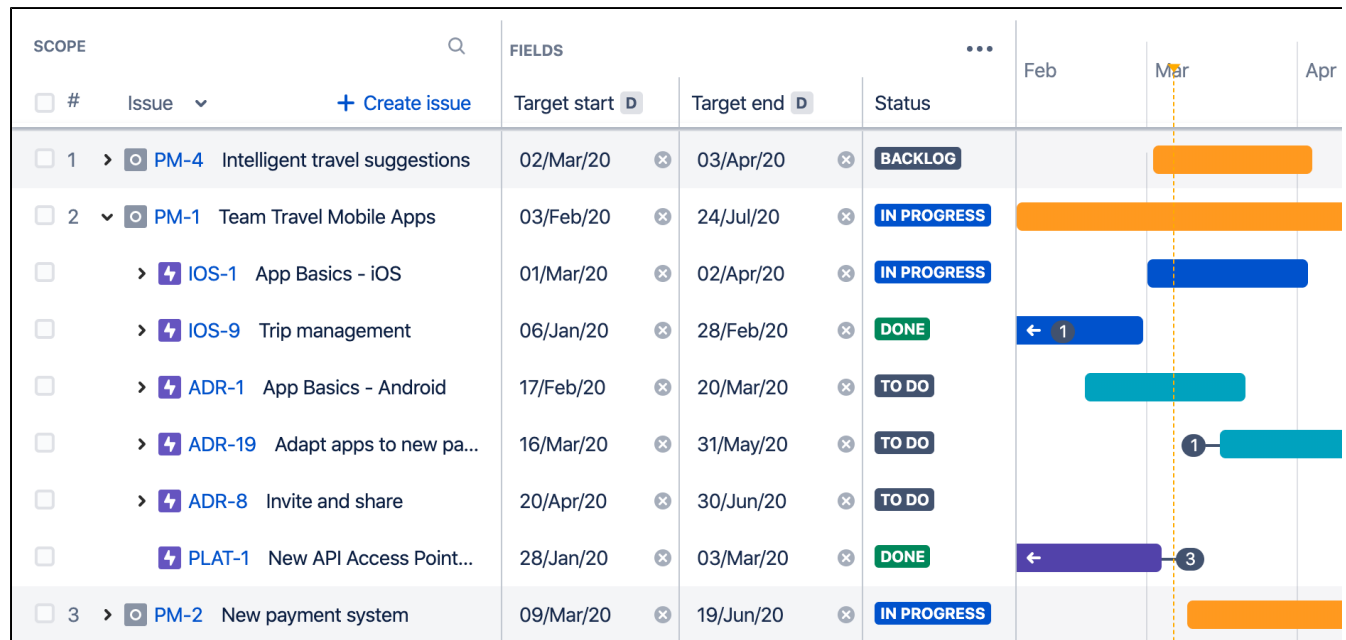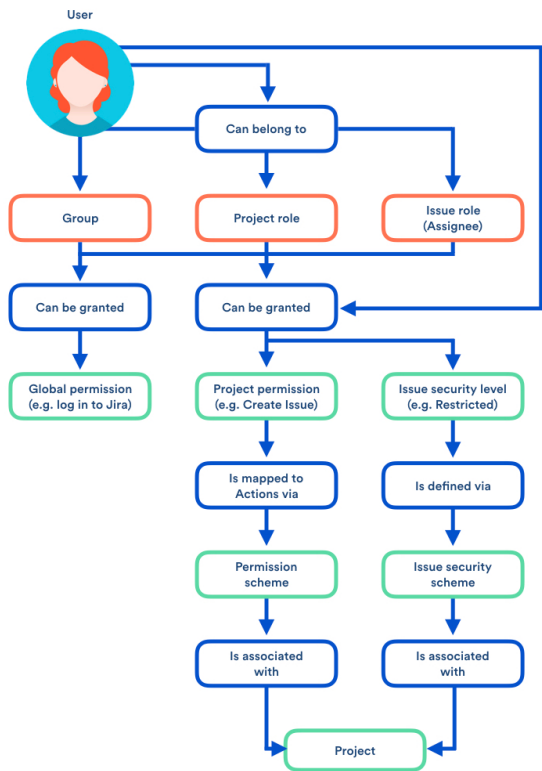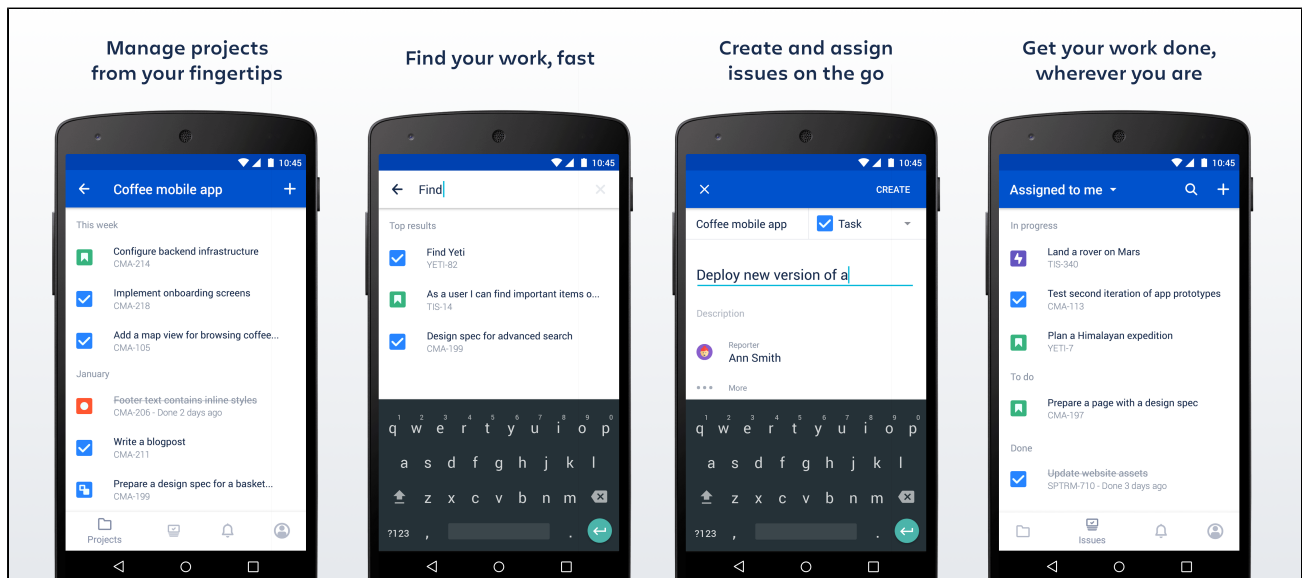Mozilla/5.0 (<system-information>) <platform> (<platform-details>) Safari/<technical-version>
AtlassianMobileApp
```

Example

```
Mozilla/5.0 (iPhone; CPU iPhone OS 14_7 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Safari
/604.1 AtlassianMobileApp
```

Android User-Agent

Format

```
Mozilla/5.0 (<system-information>) <platform> (<platform-details>) AtlassianMobileApp
```

Example

```
Mozilla/5.0 (Linux; Android 9; Mi A1 Build/PKQ1.180917.001) AppleWebKit/537.36 (KHTML, like Gecko)
AtlassianMobileApp
```

These headers are used in iOS app version 1.27.0 and Android app version 0.21.6 and later.
Mobile Device Management (MDM)

You can distribute the Jira Data Center and 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 you've 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 about push notifications, see Push notifications.
Mobile apps rely only on the 401 response status code to recognize a session timeout event. If your environment has a proxy, make sure that it doesn't override the response status code (e.g. in case of ADFS you may run into a redirect to login.microsoft.com, a 3xx status code, when your session expires). Otherwise, you might see errors like `No connection` or `Can't show the issue/page` when your session expires and should log in again explicitly to resolve them.

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
- Spinner hangs on the Notifications page
- 'Can't show the site list' 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 logging in with Azure SSO in Jira
- Can't log in to Jira mobile app with SSO/AD FS due to Windows Integrated Authentication misconfiguration

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

Invite your team to use the app

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

Head to Jira Data Center and Server mobile app to 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 instance is 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 Android 5.1 (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 Data Center and Server app to people in your organization using your MDM solution. This allows you to:

- Deploy the Jira Data Center and Server app to company-approved iOS and Android devices.
- Apply security policies to require Face ID, Touch ID, or passcode to unlock the app, or prevent users from copying, pasting, or sharing text. Requires Jira Data Center and Server iOS app version 1.29.0 or later.
- 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 Data Center and Server iOS app version 1.3.2 / Android app version 0.7.5 or later.

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

Read more about AppConfig and what we currently support in our AppConfig Technical Capabilities white 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.

Apply security policies (iOS only)

> ⓘ The keys below are only applicable to iOS because Android supports security policies by MDM providers out of the box.

Use security policies to control actions like copying, pasting, sharing text, and unlocking the app. Requires Jira Data Center and Server iOS app version 1.29.0 or later.

Key	Description
Prevent cut, copy, and paste	
`atlassian.devicePolicyClipboardRestriction`	**Value type:** Integer (1 or 0 any other value will result in an error) **Description:** Set this value to `1` to restrict clipboard use. This prevents users from cutting, copying, and pasting data from the app to other apps. Set this value to 0 to allow clipboard use. **Note:** Users will be able to copy and paste between Jira and Confluence Data Center or Cloud apps if those apps also set this value to 1.

Prevent sharing text to other devices	
`atlassian.devicePolicyDataExportRestricted`	**Value type:** Boolean (true/false) **Description:** Set this value to `true` to prevent text data from being shared to other devices via the iOS share sheet.

Require Face ID, Touch ID, or password to unlock the app	
`atlassian.devicePolicyLocalAuthRequired`	**Value type:** Boolean (true/false) **Description:** Set this value to `true` to require users to unlock the app using Touch ID, Face ID, or a passcode.
`atlassian.devicePolicyLocalAuthTimeout`	**Value type:** Integer (number greater than zero) **Description:** Enter a number in seconds (for example `100`) to define the time that the app can be open but not in use (in the background), before a passcode is required again. If you don't set a value, the default timeout is 120 seconds.

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 recommend you access the Jira Server 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 you're having problems with logging in because your proxy server blocks any unauthenticated requests. When set to true, the mobile app won't 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 you're 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 you're 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 you're not using the app. Tap the notification, and be taken straight into the app.

Here's 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 don't, your settings will still apply to in-app notifications.

Choosing what you're notified about

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 • New assignee • New comment • Issue transition
Watching	Issues you're watching	
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 thing—we've got you covered.

> ⓘ **More accessibility improvements!**
>
> Accessibility in Jira is not a one-time effort—we're constantly reviewing our VPAT report and fix accessibility issues in almost every Jira version. For more info, see Accessibility improvements in Jira.

Changing your accessibility settings

You can personalize your accessibility settings to make it easier to work with Jira. To change the settings, click your user avatar and select **Accessibility**. You can choose the following options:

- ■ **Underlined links**
  This option underlines all hyperlinks around Jira to make them more visible.

- ■ **Patterns on issue statuses**
  This options adds unique patterns to issue statuses to distinguish between them more easily.

■ **Increased text spacing**
This option increases text spacing between characters, words, lines of text, and paragraphs.

■ **Background in subtle buttons**
This options adds a gray background to subtle buttons (normally displayed on hover) to make them more prominent.

For app developers

If you're an app developer, you can make your app add new options to accessibility settings. For more info, see Accessibility for app developers.

Accessibility improvements in Jira

Making our products and services accessible to all individuals is part of our company values, and our design guidelines have been created with accessibility in mind. We've audited the Jira applications to make sure they're compliant with Section 508 and WCAG 2.0 AA. The result of this audit is the Jira Server and Data Center VPAT report.

What's a VPAT report?

VPAT, or Voluntary Product Accessibility Template, is a document that evaluates how accessible Jira is according to the industry standards and requirements, contained in Section 508. In short, it shows areas and issues we need to fix to make Jira fully accessible. For details, see Jira Server and Data Center VPAT.

Fixed issues

Here's a list of the most important issues we've fixed so far. We'll keep this list updated with every new release.

Item	Jira version
Improvements Accessibility improvements to issue search and issue view, for example: • Improved accessibility of layout and headings structure in issue search • Fixed tab order for Attachments, Links, Subtasks in issue view • Improved file attachment experience: announcing the file name when canceling an attachment, confirming successful attachment with a flag • Added label for the Share issue button Further improvements to the Create issue journey: • Improved accessibility of the Configure fields feature	8.19
Improvements Accessibility improvements to issue creation, for example: • Improved dialog and dropdown navigation • Easier user selection in issue fields • Clearer error messaging	8.18
Improvements • Accessibility improvements when issue creation	8.14

Improvements Upgraded Atlassian User Interface (AUI) to version 9, with multiple improvements to: • main header and main navigation • focus management • page layout (on selected pages) • layered elements (dropdowns, dialogs—on selected pages) **Accessibility settings** New accessibility setting: • Add background to subtle buttons	8.12
Accessibility settings New accessibility setting: • Increase text spacing	8.9
Accessibility settings New accessibility settings: • Enable underlined links • Add unique patterns to issue statuses	8.8
Improvements Continuous accessibility improvements to Jira UI in various areas, for example: • Added/improved alternative text for images • Improved the Jira code to work better with assistive technologies • Added labels/descriptions to form fields • Improved contrast of text • Improved visibility of the focus indicator	8.3+
Priority icons A set of new priority icons with distinguishable colors and looks.	8.0
Creation of the VPAT report It all starts with the first step—an audit that can tell us where do we need to improve.	7.13

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 Atlassian Community.
2. Raise a support request*.

Something isn't working

1. Check the Jira Software knowledge base at https://confluence.atlassian.com/display/JIRAKB.
2. Search Atlassian Community.
3. Raise a support request*.

If you've identified a bug but don't need further assistance, raise a bug report. (https://jira.atlassian.com)

I don't like the way something works

Raise a suggestion. (https://jira.atlassian.com)

Something else?

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

 * Tip: If you are the **Jira system administrator**, you have a number of additional support tools available. See *Raising support requests as an administrator* for details.

About our help resources

Atlassian Support

Our support team handles support requests that are raised in our support system. You need to log in using your Atlassian account before you can raise support requests.

For information on our general support policies, including support availability, SLAs, bugfixes, and more, see Atlassian Support Offerings. Note, you'll find anything security-related at Security @ Atlassian.

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 Atlassian Community, instead of having to raise a support request. This is also your best avenue for help if:

- you are using an unsupported Jira instance or an unsupported Jira platform,
- you are trying to perform an unsupported operation, or
- you are developing an plugin for Jira Software.

Jira Software knowledge base

If there are known issues with a version after it has been released, the problems will be documented as articles in our knowledge base.

Atlassian issue tracker

Our official issue tracker records our backlog of bugs, suggestions, and other changes. This is open for the public to see. If you log in with your Atlassian account, you will be able to create issues, comment on issues, vote on issues, watch issues, and more.

Tip: Before you create an issue, search the existing issues to see if a similar issue has already been created.

Automate your project

> (i) Automation is now part of Jira Software Data Center. To start using it in your projects, upgrade to Jira Software Data Center 9.0 or later. Can't upgrade yet? Download the equivalent Marketplace app free of charge.

Automation lets you set up rules to automate repetitive tasks around Jira, such as closing stale issues, auto-assigning issues to the right people, or sending notifications to your teammates.

Create a rule

Each rule is made up of a trigger that kicks off a rule, conditions that refine the rule, and actions which execute it.

To create a rule:

1. In your project, go to **Project settings**.
2. Select **Automation**.
3. Select **Create rule**. You'll then add triggers, conditions, and actions.

Read more about the rule components below.

1. Trigger

Every rule starts with a trigger. You can choose from many triggers including issue transitioned, scheduled trigger, or issue created. If you wish, you can customize your trigger before moving on to your next component and completing your rule. Learn more about triggers

2. Condition

Conditions allow you to narrow the scope of the rule. For example, only escalate an issue if it's high priority. Rules don't have to have conditions, but if you want to include them, they can be placed anywhere in the rule chain. If one condition fails, the rule will stop running. Learn more about conditions

3. Actions

Actions run once a rule is triggered and all conditions are met. You can perform many tasks, such as editing issues, sending notifications, creating sub-tasks, or adding audit messages. Learn more about actions

Use automation at scale

To give you an overview of what you can achieve with automation, we've prepared three examples of rules you can use to make the work go smoother.

Keep your board and instance clean

To make sure no ad hoc tasks disturb your team's workflow, you can set up a rule that will handle any issues that aren't epics or stories by assigning them to the right people.

The rule consists of:

- A trigger: An issue is created.
- A condition: The issue is a bug or a task.
- Actions: Assign the issue to an automatically selected member of the special-task-force group using the balanced workload policy, add watchers to set supervisors for the issue, and then transition the issue to the "To do" state.

Keep the work flowing

With many active issues comes the hard task of keeping track of them. Luckily, you can set up a rule that will find any issues that haven't been updated for a while and send out a reminder to the assignee.

This rule is built on top of the scheduler that allows it to check for issues selected in a JQL query every 4 hours. Since there is a JQL result limit in Automation for Jira, this rule runs more than once a day to fully process the JQL results even at a large scale.

The rule consists of:

- A trigger: A scheduled operation, configured to run every 4 hours to process the results of a JQL query. The query finds issues (bugs and tasks) that haven't been updated for 7 days and aren't resolved.
- A condition: An if-else condition checks if the issues have an assignee and proceeds to an action based on the result.
- Actions: If the issue has an assignee, an email is sent asking them to update the status of the issue. If there is no assignee, an email is sent to the project supervisor asking them to assign someone to the issue.

Automate repetitive tasks

For every release, there are some activities that are repetitive and thus, unnecessarily time-consuming. For example, updating your documentation or building an artifact. With a rule that helps automate those processes, your team can focus on what's really important.

The rule consists of:

1. A trigger: The rule is executed when a new version is created in the project. There is also a version name constraint that causes the rule to only be valid for final releases whose version string contains 3 numbers separated by dots (e.g. 1.0.0).
2. Actions: Create new stories in the project—the first one in the "Product documentation" epic, the second one in the "Product releases" epic.