



Jira Essentials

Copyright ©2020 Atlassian Pty Ltd. All rights reserved.

Atlassian®, Atlassian Bamboo®, Atlassian Clover®, Atlassian Confluence®, Atlassian Crowd®, Atlassian Crucible®, Atlassian Marketplace®, Bitbucket®, Confluence®, Confluence Questions®, Fisheye®, Hipchat®, Jira®, Jira Portfolio®, Jira Service Desk®, Sourcetree®, Statuspage®, Stride™, Trello®, ▲ ATASSIAN®, and ▲® are trademarks of Atlassian Pty Ltd. All other trademarks and trade names are the property of their respective owners. Information in this document is subject to change without notice. Websites, names, and data used as examples in this document are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording by any information or retrieval system, for any purpose, without prior written permission of Atlassian Pty Ltd. Address inquiries to Atlassian Pty Ltd, Level 6, 341 George Street, Sydney, NSW 2000 Australia.

27 Apr 2020 – Cloud and Server



1

Course Overview



What will you learn?



- Describe Jira
- Identify the differences between kanban and scrum projects
- Use Jira to organize, find and report on your work
- Configure Jira to match your team's processes



To succeed here, you need to have

- No knowledge of Jira is assumed



Schedule



1	Course Overview
2	Jira Overview
3	Project Boards
4	Enrich Issues
5	Kanban Projects
6	Scrum Projects



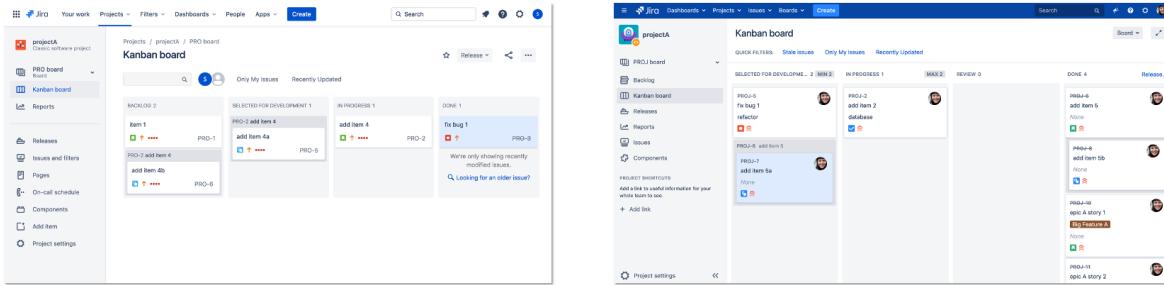
Schedule



7	Quick Search and Basic Search
8	JQL
9	Filters
10	Epics
11	Dashboards
12	Putting it all Together



Jira Cloud vs Jira Server/Data Center



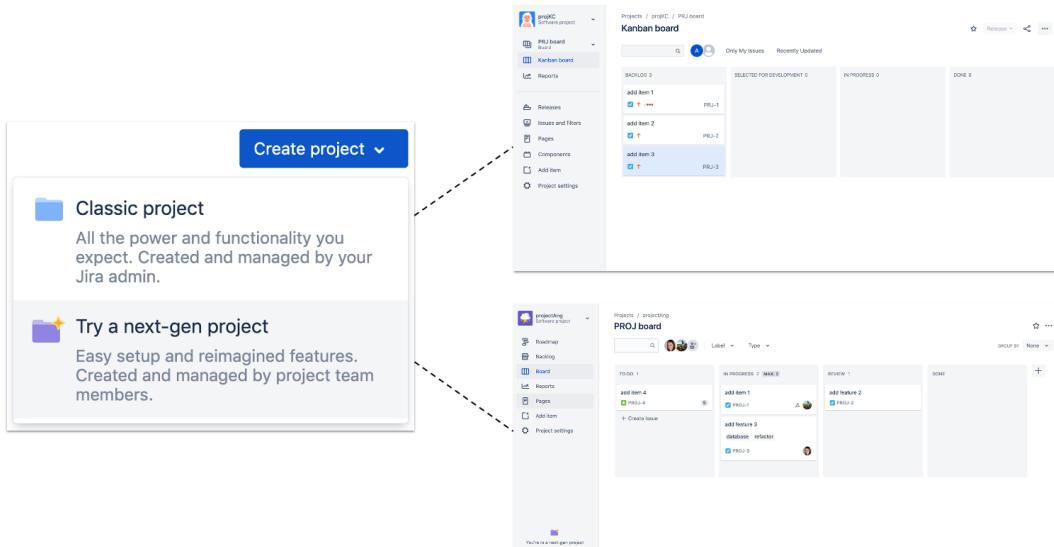
Cloud

Server /
Data Center



There are two main ways that a company can choose to run their Jira instance. Using Cloud, Atlassian does the work of hosting and maintaining the Jira software, so that the company can focus on their value-added activities. This is much like a company that chooses to host their email using an external service such as Google Gmail. If a company wants to host their own Jira instance, they could choose the Server or Data Center version. They can host their Jira instance in the cloud, for example using Amazon Web Services (AWS), or they can host their Jira instance in their own data center. The Server version of Jira is hosted on a single node. The Data Center version is hosted on multiple nodes, leveraging the reliability benefits of a distributed architecture.

Jira Cloud- classic vs. next-gen projects



In the Cloud version of Jira, you are given a choice of creating a classic project or a next-gen project.

Classic projects are traditional Jira projects closely resembling Server/Data Center projects. In general, for now they are more mature and have more functionality than next-gen projects. They have more configuration capabilities, and configurations can be shared among multiple projects. They are mostly created and managed by Jira administrators.

Next-gen projects are newer to Jira, and capabilities are being added by Atlassian engineers in an agile way. In general, for now you will see more changes to next-gen projects than to classic projects. While there are many similarities, Next-gen projects have a different perspective on managing the projects. Instead of Jira administrators mainly creating and configuring projects, next-gen projects can be easily created and configured by the members of the project team. The configuration applies only to that project, so team members don't have to worry that they are changing the configuration of other projects.

The type of project that you create is up to you and your organizations. Some organizations may choose to only use classic projects and keep better control over the project configuration. Some organizations may prefer the ease and flexibility of next-gen projects. Some organizations may allow the project teams to choose.

Tasks

Log in to Jira

- Decide if you want to do the Cloud or Server version of the labs
- Log in to Jira



2

Jira Overview



What will you learn?



- Describe Jira
- Create a Jira project
- Create a Jira issue
- Use a project board
- Identify Jira user types



Topics

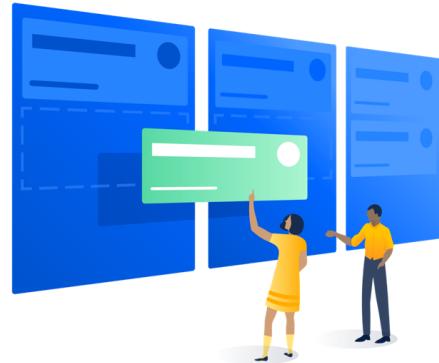
Jira overview

Projects, issues and boards



What is Jira?

- A tool used to help teams perform, visualize and manage work
- Models the team's current processes/workflows

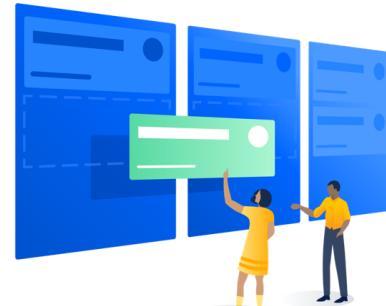


Jira is a tool that teams can use to help facilitate agile work. It helps the team perform their work, visualize work and manage the work using things like boards, reports and dashboards. A team can use Jira to help make better decisions throughout the life of the project.

Jira is very configurable, allowing the team's current desired processes or workflows to be modeled and used to accomplish the team's work. You want the tool to conform to the processes that your team desires, not be constricted by the capabilities of the tool. Because agile process are always open to improvement, you want your tool to be adaptable as your team learns. A solid understanding of Jira's capabilities helps the team work with confidence as they continue to improve.

Why Jira?

- Leverage project management technology, allowing teams to focus on their work
- Facilitates planning, prioritizing, organizing and completing work
- Visualizes work using project boards, reports and dashboards
- Facilitates team communication



Jira provides administrative project management capabilities out of the box, so teams can focus on the work that differentiates them. For example, Jira reports are automatically created and updated as the team performs its work.

Jira helps the team focus on its work, facilitating planning, prioritizing, organizing and completing the work of the team.

A key agile/lean principle is to visualize work. Jira provides work visualization out of the box using things like project boards, reports and dashboards.

Jira facilitates team communication in ways such as visualizing the work of the team, informing the team of the planned work, allowing teams to discuss work items, and by notifying team members when the work items change.

Topics

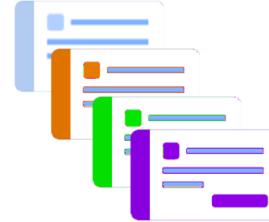
Jira overview

Projects, issues, boards and user types



What is a Jira issue?

- An item of work (work item) identified by the team
- An issue has an associated type (for example, story, task, bug)
- The details of the issues are known as *fields*



Issues



A Jira issue is the name of an item of work that has been identified by the team. The term “issue” comes from Jira’s historic roots as bug, or issue tracking software.

Every issue has an associated issue type. You will work with these throughout the course, but issue types include stories, tasks, bugs, epics and custom issue types. Each issue type can have unique screens and workflows associated with them.

An issue can contain a lot of information. This information is broken down into fields. Fields include the name of the issue, its unique identifier, its description, comments, date of creation, current assignee and many others. Custom fields can also be created to match your team’s desired business processes.

What is a Jira project?

- A collection of related issues
- A team “to do” list
- Can have a fixed end date or be an ongoing project
- A project has an associated type (for example, kanban, scrum)



A Jira project is a collection of related issues. A project is a way to organize work. The issues can be related in any way that the team desires.

You can think of the issues of a project as the team’s “to do” list. The issues can be in different states, such as not ready for development, ready for development, in progress and done. Jira therefore contains a record of the team’s work, allowing the team to do things like report on their work.

A Jira project is not necessarily a project in the traditional sense, which usually has a start and end date. The term “project” is used loosely to include work that has no planned end date. For example, a product that is planned to continuously be improved, with no plan to ever stop work on it.

Projects have an associated type, depending on how the team wants to accomplish their work. When you create a project, you select its type, such as kanban or scrum. We will discuss these later in the course. Over time, the team can configure their project to use a custom agile methodology or framework.

Issue key



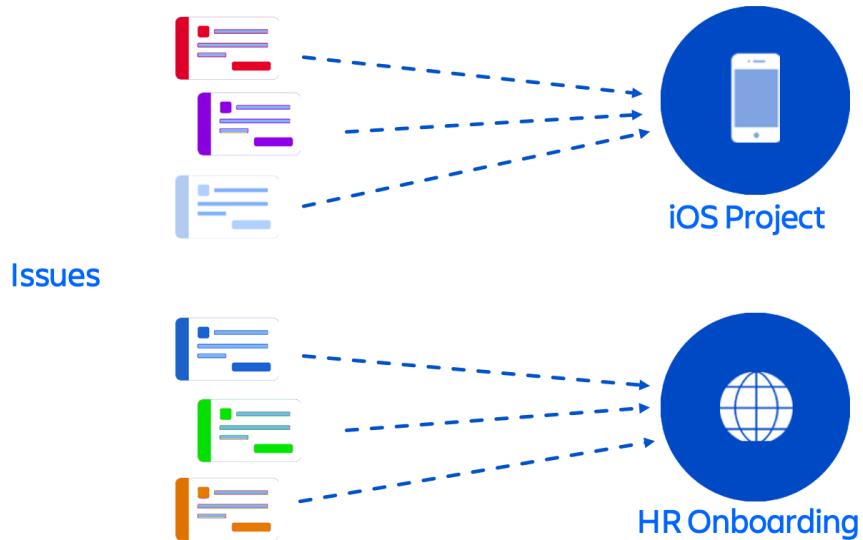
Jira automatically assigns a unique *issue key* to created issues

`<issue_key> = <project_key>-<issue_number>`



Jira automatically assigns a unique issue key to each issue. The letters before the dash represent a unique identifier for the project. This is called the *project key*. The issue number in the project follows the dash. Issue key values are unique to the Jira account. To ensure this, you can not have two projects with the same project key.

Each issue belongs to one project



Every issue in Jira is unique and belongs to just one project. The issues are the project's work items. For instance, the HR Onboarding project may have issues such as create email account, supply laptop and assign desk. The iOS project may have issues such as code login button, create help screen and configure iCloud.

No issue can belong to multiple projects.

What is a project board?

- A two-dimensional “to do” list
- A way to visualize issues
- A visualization of the team’s process/workflow
- Displays issues as *cards*



A project board is a two-dimensional view of the work to be done by the team. The two dimensions become more important than with a personal to do list because the work items can be in multiple states and be worked on by multiple team members. Project boards visualize the work process of the team, and can be physical boards (such as sticky notes in columns) or digital boards (such as Jira boards).

On a Jira board, each issue is shown as a card, which displays a convenient subset of the issues’ fields for easy visualization. The fields displayed on a card can be configured to match the team’s desires.

Jira user types



There are three main types of Jira users. A *Jira administrator* configures the Jira instance for all users. They in general know the most about the technical capabilities of Jira and can set policies for the entire company to use with their Jira projects. The changes that they can make can affect multiple projects, so they must be very knowledgeable of Jira. A *Jira project administrator* can configure a specific project to match the team's desired process. Jira project administrators work closely with the agile team to understand the desired process, and works with the Jira administrator when they do not have permissions to perform some of their desired Jira configuration.

A team member uses Jira to work on projects.

In general, a company has a few Jira administrators, more Jira project administrators, and even more team members.

Takeaways



- Jira is a tool teams use to manage and visualize the work of a project
- A Jira issue is an item of work identified by the team
- Project boards visualize a team's work
- The main types of Jira users are Jira administrators, Jira project administrators and team members



Tasks

Create a project

- Create a project
- Create issues



3

Visualize Work Using Project Boards



What will you learn?



- Describe the importance of visualizing work
- Describe common workflows
- Differentiate Jira boards and workflows
- Describe the purpose of an issue's *status* field
- Configure board columns



Topics

Visualizing work

Workflows

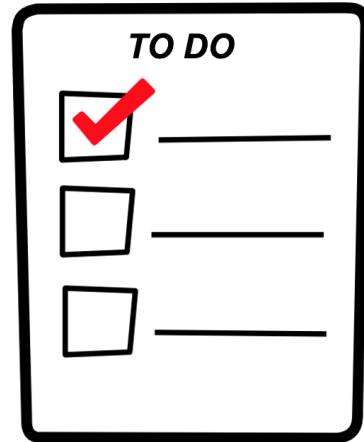
Jira boards and workflows

Configuring board columns



Visualizing work: a “to do” list

- Reminds you
- Focuses you
- Sets priorities
- Tracks progress



A to do list is a classic example of a tool used to visualize your work. Even though a to do list is very simple to use, it has some valuable characteristics.

One is that it visually reminds you of the work that you need to do. We all have a tendency to forget things if we don't write them down, especially if there are a lot of items. A to do list helps prevent you from forgetting things.

A to do list focuses you, because you can concentrate on doing only the things on the list.

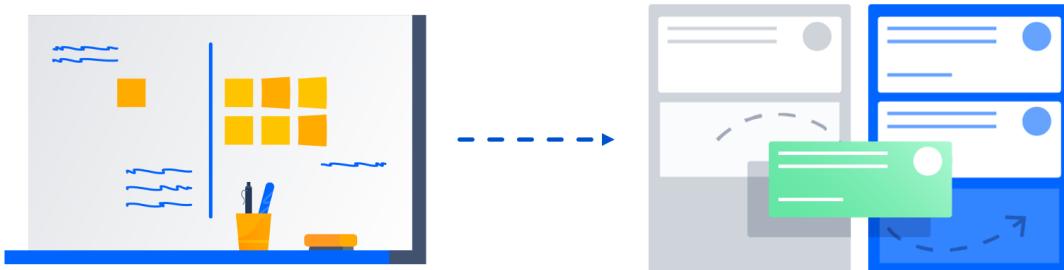
If you want, you can prioritize the work items on your to do list, simply by changing their order.

As you check off your work items, you are tracking your progress. This is usually quite rewarding.

We will see that this simple example of a to do list contains a lot of the building blocks related to visualizing the work of agile teams.

Visualizing work: a board

- A principle of agile projects is to "visualize work"
- A board is an agile tool used to visualize and manage work



A principle of agile projects is to visualize work. A board is an agile tool used to help visualize and manage the work of the team. Depending on the circumstances, a board may also be referred to as a task board, a project board, a kanban board or a scrum board.

A board can be a physical board like a whiteboard with sticky notes, or can be based in software. The board shown here is in Jira.

Depending on which template that you choose when creating the project, a board can be automatically created for you.

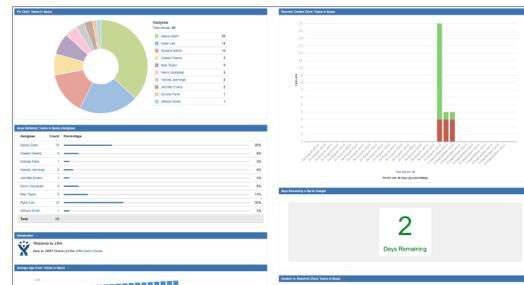
You can see that the board contains columns. Each column can contain issues, which are the work items for the project.

You can see that a board is like a two-dimensional to do list. The extra dimension is used to allow a work item to go through multiple steps by multiple team members before it is finished. Agile teams commonly break up the project into manageable issues and work on them in steps like this.

Visualizing work: reports and dashboards



Reports



Dashboards



We saw that project boards are used to visualize the work of the team. In Jira, reports and dashboards are automatically created and updated for you. This is another great way to visualize the team's work, especially for identifying problems.

Why visualize work?

- **To easily see the work of the project**
 - Allows anyone to see the (true) current state of the project
 - Organizes and focuses the team
 - Only work on tasks on the board
- **To manage things**
 - Easy to add and prioritize the work of the project
 - Easy to update work items
- **To improve the team's way of working**
 - Can visually identify problems



Why is visualizing work an important principle of agile? Like a to do list, visualizing work with tools such as a board makes things easier, better and more rewarding. Visualizing work allows everyone to see the current state of the work of the project. With a board, you can easily see which items are done, which items are in progress, and which items have not been started. Boards are very transparent in that they show the true state of the project, not only to the project team, but to any stakeholders who have access to the board.

Visualizing work also organizes and focuses the team. The team should only be working on the issues on the board, and the next issue to be worked on is very obvious.

Visualizing work allows you to manage things. For example, the team can easily add items to the board, modify existing items or change their priority. As the team works on a work item, the item can be updated to show the progress.

Visualizing work can also improve the team's way of working. A key principle of agile projects is to continuously improve, not only the product, but the way that the team works on the product. A board can help the team visually identify problems or bottlenecks with the process. This highlights areas for the team to focus on improving.

Topics

Visualizing work

Workflows

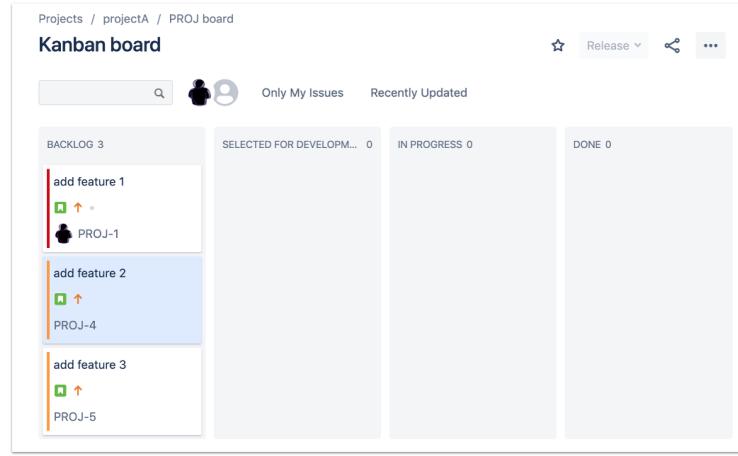
Jira boards and workflows

Configuring board columns



Workflows

- The set of columns of a board represent a **workflow** (or process) for completing the work of an issue
- Workflows are broken down into **statuses** (or steps)



The set of columns of a board represent a workflow for completing an issue.

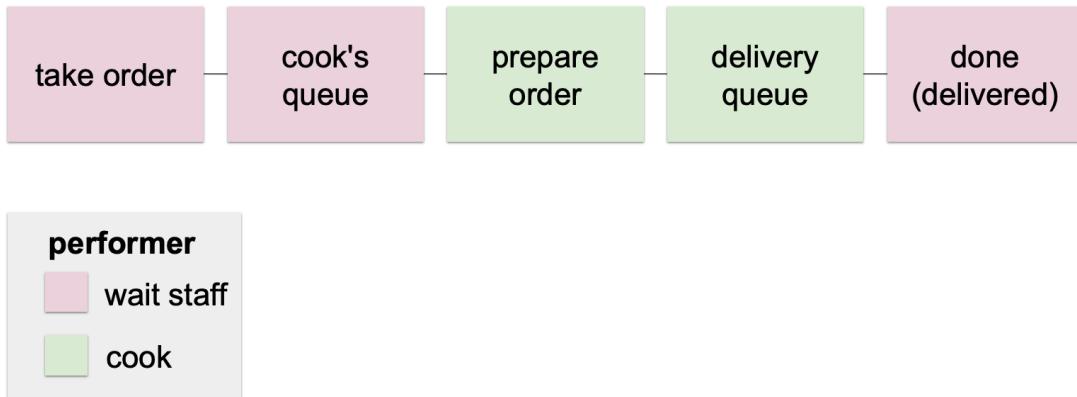
Workflows are used to model the processes involved in the project. Even though a workflow technically can be considered a model of a process, you often will find the terms workflow, process, business process and value stream used interchangeably. They all represent breaking down the work into a series of steps.

Each column in a board usually represents a single step in the workflow. These steps may also be called statuses, states or stages.

You can see that workflows and boards are closely related. The board visualizes the workflow.

The board shown here has four statuses in the workflow. These statuses happen to have the same name as the columns in the board. The statuses are backlog, selected for development, in progress and done.

Example workflow: restaurant order and delivery



Let's take a simple process and model it as a workflow. We will model the process of ordering and delivering a customer's food at a restaurant.

The first step of the workflow is for the wait staff to take the order from the customer. In the second step, the wait staff adds the order to the cook's queue. It's placed in a queue because the cook only can start the order when they have bandwidth.

In the third step, the cook takes the order from the incoming queue and begins to prepare the order.

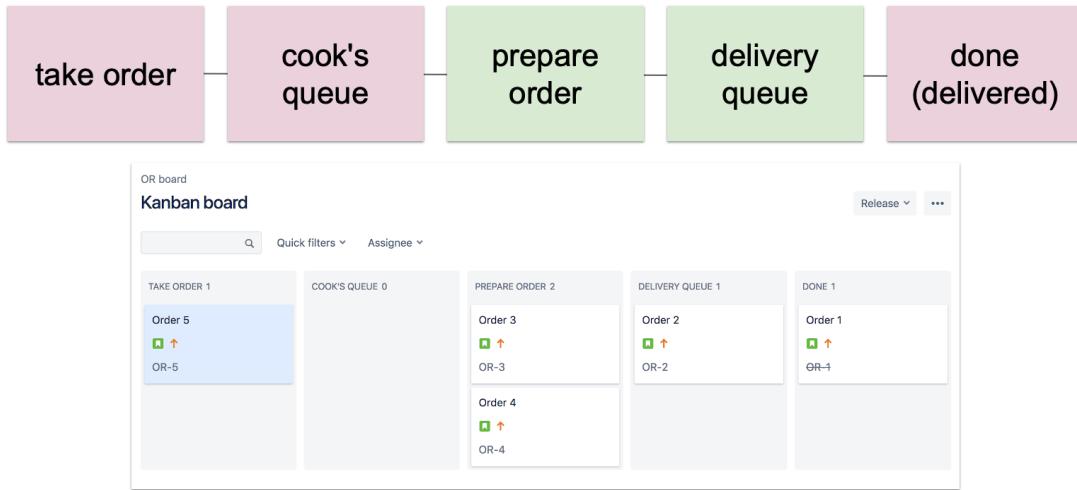
In the fourth step, the cook has finished the preparation and adds the order to the delivery queue. It's placed in a queue because the cook usually doesn't directly deliver the order to the customer and the wait staff might not be immediately available.

In the final step of the workflow, the wait staff delivers the order to the customer. Even though this is a simplified example of a workflow, the same basic idea of breaking down the work of the project into steps applies. One of the benefits of breaking down the workflow into steps is that the work of the steps can be done by different performers, allowing the process to scale to large teams.

You can model any process that you want using this simple block diagram approach. All it takes is a pencil and paper.

Boards vs. workflows

- A team works using a board
- The board's structure is defined by an underlying workflow



We can see that boards and workflows are closely related. A team works using a board. The board's structure is defined by an underlying workflow. This example shows the restaurant workflow that we just defined. Below that, we have created a Jira project with columns of the board matching the workflow. We will discuss configuring boards later.

Topics

Visualizing work

Workflows

Jira boards and workflows

Configuring board columns



How are boards created?

- Automatically
- Create additional boards at any time

The screenshot shows a Jira Kanban board interface. At the top, it says 'Projects / projectA / PROJ board'. The board title is 'Kanban board'. There are three columns: 'BACKLOG 3', 'SELECTED FOR DEVELOPM...', and 'IN PROGRESS 0'. In the 'BACKLOG' column, there are two cards: 'add feature 1' and 'add feature 2'. The 'add feature 1' card is associated with 'PROJ-1'. The 'add feature 2' card is associated with 'PROJ-4'. A context menu is open on the right side of the board, listing options: 'Board settings', 'Create board', 'Hide menus', 'Show detail view', 'Open issues in sidebar', 'Expand all swimlanes', 'Collapse all swimlanes', 'Hide epic labels', and 'Print cards'. The 'Create board' option is highlighted.

Boards are automatically created when you create a project using kanban or scrum templates. In this example, we have created a kanban project. You can create additional boards at any time. A single project can have multiple boards, and a single board can contain the issues of multiple projects. (Currently, you can not create extra boards with Cloud next-gen projects.)

An issue's status field

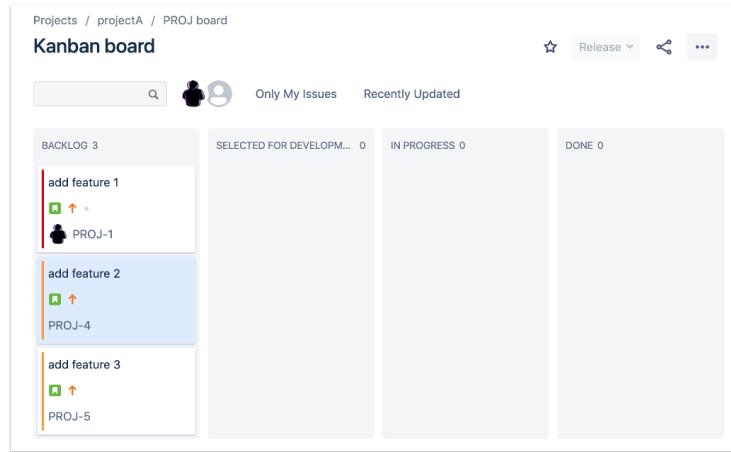
- Every project automatically has an associated workflow
- The **status** field for each issue must be set to one of the workflow's statuses



In Jira, every project that is created automatically has an associated workflow. The status field for each issue must be set to one of the workflow's statuses. In this example, we are viewing the details of an issue with an issue key of PROJ-1. You can see that there is a Status field. This issue's status is currently set to Backlog, and the dropdown box shows that there are three other statuses that we can set for this issue.

Boards and status

- Boards are a view of issues arranged by *status*
- Moving an issue changes the value of its *status* field



We are now in a better position to understand boards. Boards are a view of issues arranged by status. When you move an issue to a different column in the board, you are changing the value of its status field.

Changing an issue's status: status field value



Another way to change an issue's status is in the issue details. You can change the status using the Status dropdown box. This will move the item to a new location on the board.

Topics

Visualizing work

Workflows

Jira boards and workflows

Configuring board columns

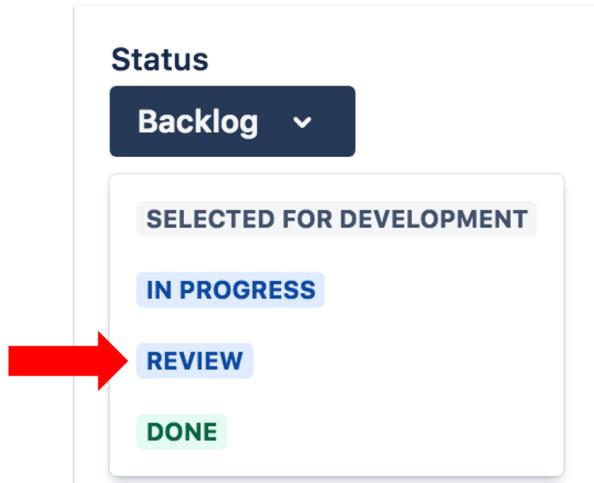


Adding a board column

The screenshot shows a Jira Kanban board for projectA / PROJ board. The board has five columns: BACKLOG 2, SELECTED FOR DEVELOPM... 0, IN PROGRESS 0, REVIEW 0 (highlighted with a red border), and DONE 1. In the BACKLOG 2 column, there are two issues: "add feature 2" and "add feature 3", both from PROJ-4. In the DONE 1 column, there is one issue: "add feature 1" from PROJ-1.

Here we have added a Review column (and its related status) to the board. Issues can now be moved to this column, exactly like moving them to the other columns.

Viewing the new status



If we view an issue's details, we can select that the Review status is now selectable.

Takeaways



- A project board is a two-dimensional way to visualize the work of a team
- In Jira, a workflow is often represented using a project board
- Project board columns usually map to the **status** field of issues
- Board columns can be added or removed to match the team's desired process



Tasks

Visualizing Work Using Project Boards

- Move issues through a workflow
- Assign an issue
- Add a Review column to the board
- Explore the difference between Jira project administrators and standard users



4

Enrich Issues



What will you learn?



- Identify ways that issues can be enriched with information
- Describe the benefits of using issue types
- Describe subtasks
- Use labels to organize issues
- Introduce integration with version control and build systems



Topics

[**Enriching issues**](#)

[Issue types](#)

[Labels](#)

[Developer integration overview](#)



Issues contain work-related information

Issue

Summary: [Check network jacks](#)

Description: Each network jack in the new building needs to be checked for signal strength.

Type:  Task Assignee: 
Helena

Priority:  Critical Reporter: 
Oliver

Status: [IN PROGRESS](#)

Comments: Helena needs the network diagram from IT.



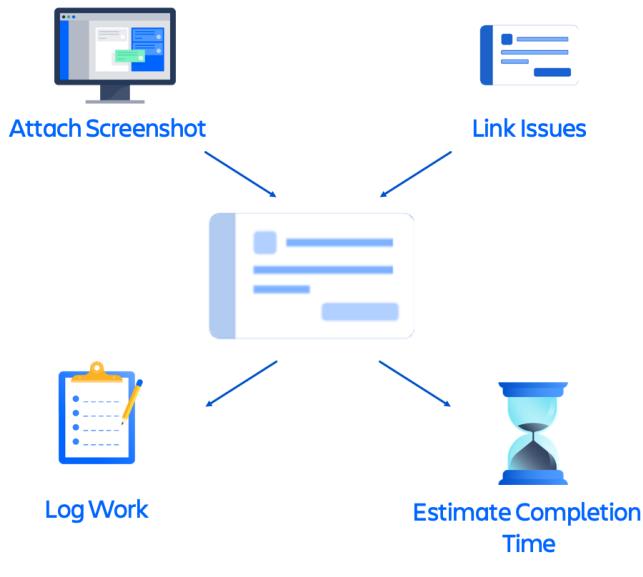
An issue in Jira can contain all the information about that particular work item in one place. This information is stored in fields.

The assignee is the person currently assigned to work on the issue. This often changes throughout the life of the issue.

The reporter is usually the person who created the issue. An issue will only have one reporter and this rarely changes.

What fields you see in an issue depends on the type of issue and how your project is configured. You can also create custom fields to match your team's desired process.

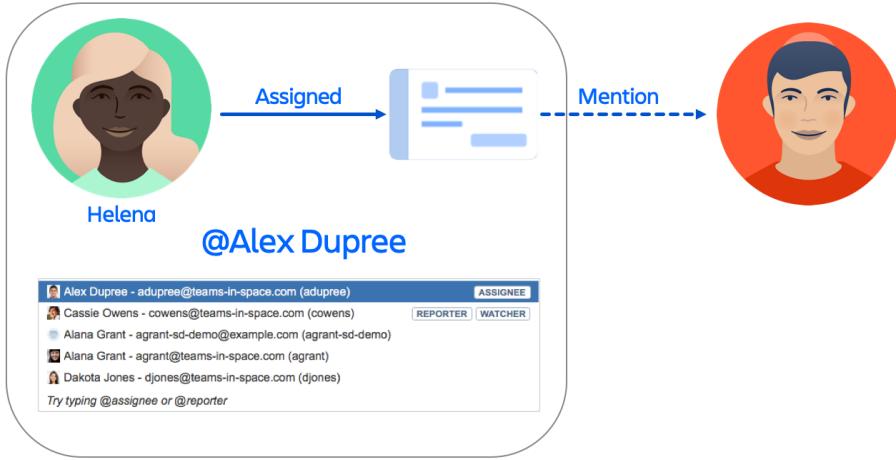
Enriching issues



Here are some other common actions you can perform to enrich issues:

- You can add **attachments** to issues, such as a **screenshot** showing an error or an **invoice** file. This keeps all the important information easily accessible for the issue.
- You can **link** an issue to another one. For example, a problem request in an IT Service Desk can be linked to the bug that caused it in Jira Software. By linking these two issues, the support engineer can easily navigate to the bug issue to see what's being done. Also, the problem request will automatically be updated when the bug is fixed.
- When you create an issue, you can estimate the **time** needed to complete it.
- As you work on an issue, you can **log the time** and the actual **work** performed on the issue. This is important for project planning, tracking and reporting.

Mention team members



You can “at mention” (@mention) team members in an issue, and they will be notified that they have been at mentioned.

By simply entering the @ character, a list of available team members will appear for you to select. Jira will then notify that team member that they have been mentioned in a particular issue.

Helena is assigned to add some new artwork to the company website. She needs to get the files from Alex. Helena enters @Al and Alex’s name appears in the list for Helena to select. Helena has now tagged Alex to the issue by using an at mention.

Topics

Enriching issues

Issue types

Labels

Developer integration overview



The issue type field

- **Epic**- a big issue that can contain issues
- **Story**- requirement from the user's perspective
- **Task**- team work item
- **Bug**- a flaw that needs to be fixed
- **Subtask**- a child of another issue

Create issue

Project*
projectA (PROA)

Issue Type*

Story

Task

Bug

Epic

A project can also use custom issue types



In Jira, an issue is a generic name for a unit of work. On your projects, there usually are different types of units of work. The issue type field is used to differentiate these different types of units of work. When you create an issue, you can choose the issue type. You can also change the issue type after creating the issue. Notice that each type has an associated icon to help easily identify the issue type.

A story is a requirement from the user's perspective.

A task is a work item that needs to be done by the team, but is not directly tied to a user requirement. An example might be upgrading the version of a product used by the team.

A bug is flaw that needs to be fixed in the product. It can be tracked with its own issue type to differentiate this work from other types of work.

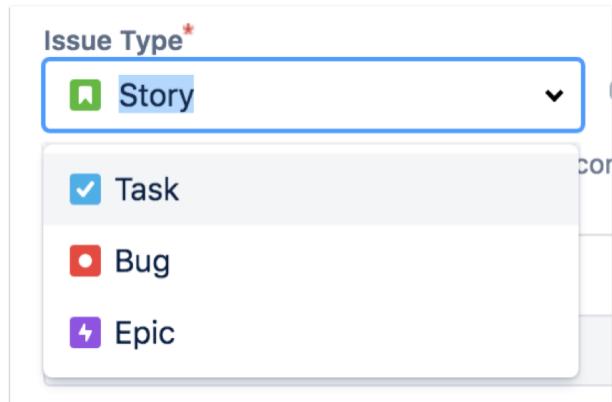
An epic is a big issue that can contain other issues.

A subtask is a child of another issue. It is used to break an issue down into specific pieces of work. When creating an issue, subtask is not shown in the issue type dropdown because subtasks must have a parent issue- they can not be created independently.

In addition to these out of the box issue types, you can create custom issue types. This provides your team the flexibility to work the way that they want to work. How the team actually defines what each issue type means is entirely up to them. Also, different projects can use different issue types.

Why issue types?

- Supports different types of work
- Each type can have different fields, screens and workflows
- Can report on types separately



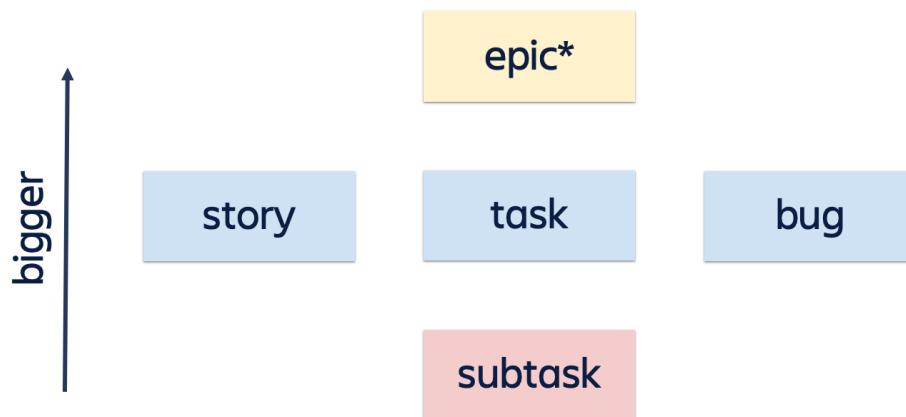
There are several reasons to use issue types with your projects.

Issue types support different types of work items. A team usually doesn't have only one type of work, and issue types allow the team to differentiate those types of work.

Each type can have different fields, screens and workflows. For example, you might want bugs to appear at the top of the project's board.

You can also report on types separately. For example, because the issues of the project have been categorized by issue type, you can easily create a report with the number of bugs fixed in the previous week.

Jira's issue type hierarchy

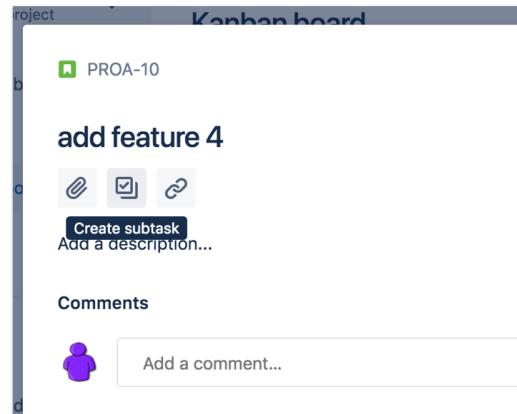


* Epics are discussed later



Subtasks

- An issue type that must have a parent issue
- Allow an issue to be broken down into individually manageable tasks
- Can be more technical than the parent issue



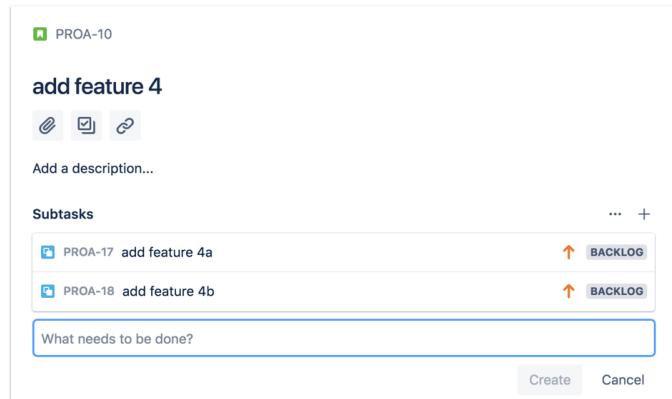
Subtasks are an issue type that must have a parent type.

Subtasks allow an issue to be broken down into individually manageable tasks. They can be assigned to different team members.

Subtasks can be more technical than the parent issue. For example, if the parent issue is a story, the story will be written in non-technical language that all team members and stakeholders understand, but the subtasks can be written for the technical person implementing the subtask.

Subtask characteristics

- Have their own issue key and field values
- Have independent workflow status



Subtasks have their own issue keys and fields. Here is a story with two subtasks. Each subtask has its own issue key and summary field value.

The subtasks have independent workflow statuses and move through boards independently.

Topics

Enriching issues

Issue types

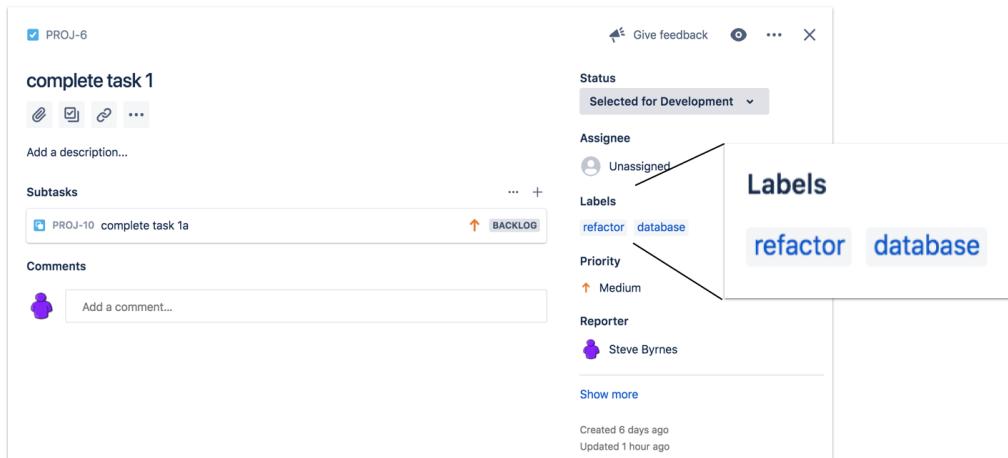
Labels

Developer integration overview



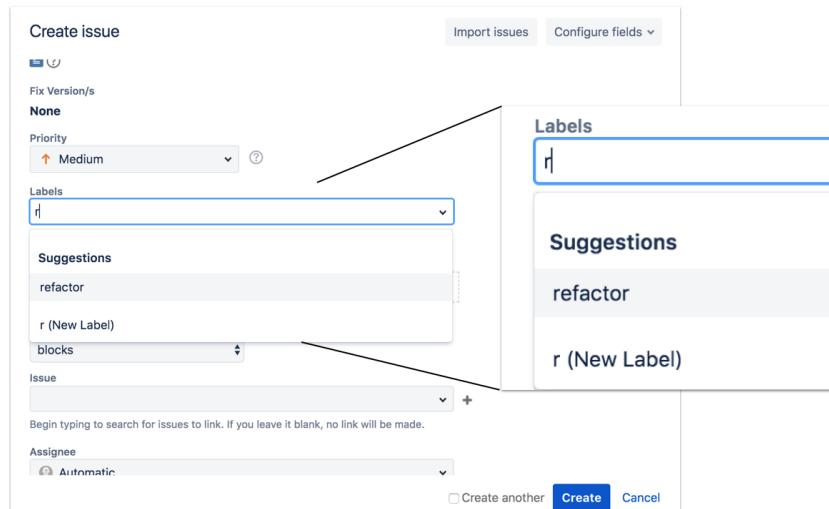
Labels

A field used to categorize and search for issues



Labels are a field used to help categorize and search for issues in any way that makes sense to the team. You can have multiple labels per issue. This issue has two labels- refactor and database.

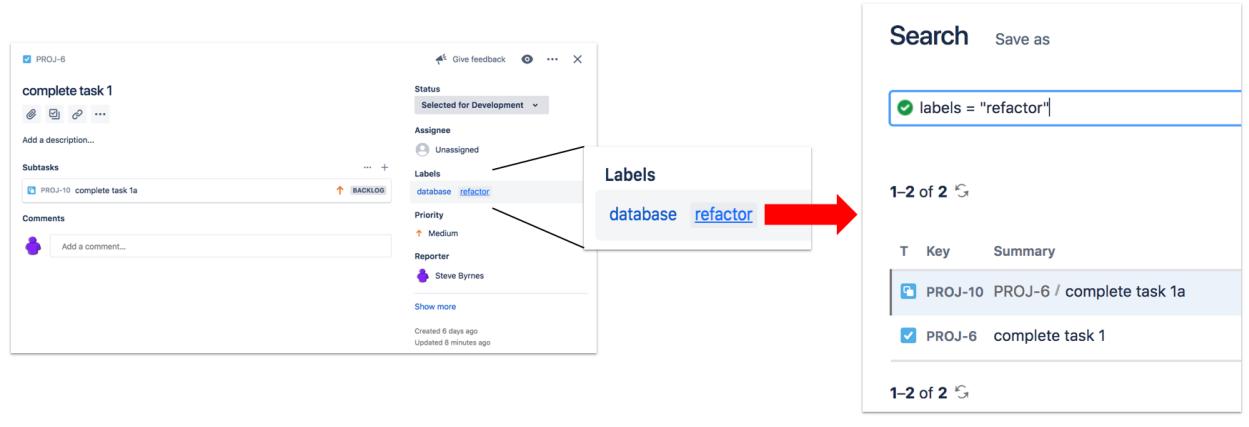
Adding or creating a label



Labels can be added as you create an issue or can be added to existing issues. Jira will suggest existing labels as you type. In this example, in the create issue screen, we typed an "r", and we see that we can select the refactor label. We could also type a new name and the label will be created.

Searching for labeled issues

Click on a label to search for all issues with this label



You can click on a label in an issue, and Jira will take you to a search page containing all issues with this label. In this case, two issues use the “refactor” label. You can also search for issues with certain labels using basic search or JQL, which we will discuss later.

Topics

Enriching issues

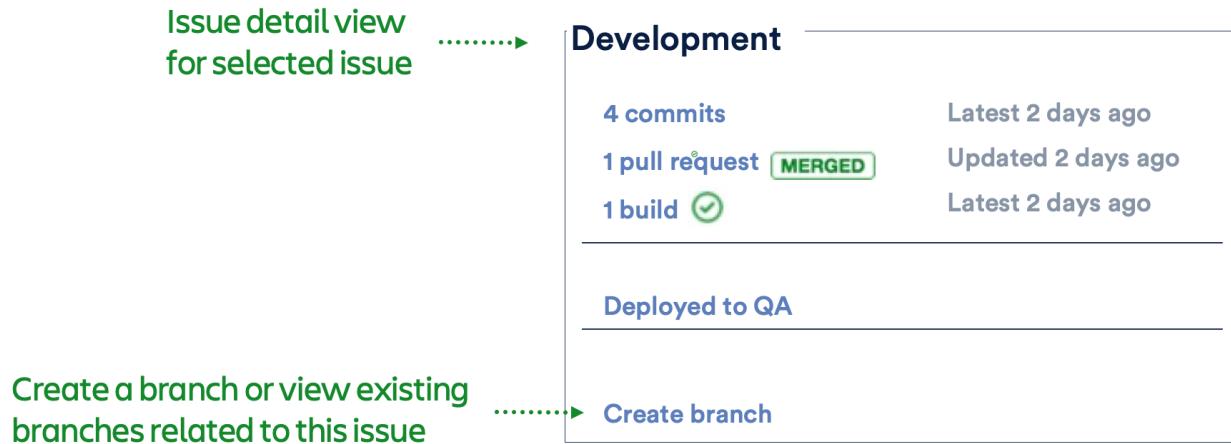
Issue types

Labels

Developer integration overview



The issue detail development panel



When Jira Software is integrated with development tools, the development panel is displayed with the issue details.

In the development panel, we see that there are four commits related to this issue, one pull request that's been successfully merged, and one successful build. We also see that this issue was included in a deployment to QA.

Integration works through the issue key

Using Commit Message

Include issue key in commit message
“Initial commit – TIS-498”

Using Branch Name

Include issue key in branch name
“Feature branch TIS-498”

For Pull Requests

Include issue key in pull request title,
or Jira can use issue key from
associated commit or branch

For Builds and Deployments

Jira uses the issue key associated with
a commit in the build



The integration is enabled through a reference to the issue key. This reference can be in a commit message or a branch name.

For pull requests, the issue key can be used in the pull request title, or the pull request can just get the reference from a commit or branch associated with the request.

Builds and deployments use the issue key reference from an associated commit.

Takeaways



- An issue contains a diverse set of fields that are used to add information to the issue
- Issues can facilitate team communication with comments and @mentions
- Issue types can have unique fields, screens and workflows
- Subtasks are children of another issue type
- Subtasks have their own issue key and field values
- Labels can be used to categorize and search for issues
- Jira can be integrated with version control and/or build systems to improve developer-related communication



Tasks

Enrich Issues

- Add information to an issue
- Use team-related issue features
- Create issue of different types
- Create subtasks
- Add labels to issues



5

Kanban Projects



What will you learn?



- Describe the kanban method
- Describe the importance of flow
- Identify the purpose of work-in-progress limits
- Identify reasons to separate the backlog from the board



Topics

Kanban method overview

Improving flow

Separate backlog



What is the kanban method?

- An agile method used to manage a continuous queue of work items
- Commonly used ideas:
 - Visualize work
 - Remove process bottlenecks to improve "flow" of value
 - Limit work in progress / small batch size
 - Pull work rather than push (where it makes sense)
 - Continuously prioritize work items

[https://en.wikipedia.org/wiki/Kanban_\(development\)](https://en.wikipedia.org/wiki/Kanban_(development))



The kanban method is an agile method commonly used to manage a continuous queue of work items. The kanban method leverages many of the ideas of the Toyota Production System, which we will discuss later in the course.

Some of the ideas used in the kanban method include:

Limit work in progress. At any given time, the team should only be working on the amount of work that it can handle sustainably. Limiting work in progress results in small batch sizes.

Remove bottlenecks to improve flow. Ideally you would like to have the team outputting a steady stream, or flow of work. The reality is that there will be bottlenecks in the process. Issues will become stuck in certain places and may pile up in other places. These can be because of a problem with the process itself, or just the reality of the complexity of the issues. Either way, the team works together to remove these bottlenecks once they are identified. Attempting to remove bottlenecks forces the team to "see the whole" process rather than just a portion of it. When removing the bottleneck, it is best to find the root cause and fix the process there.

Pull work rather than push, where it makes sense. For many steps of a process, it is better for someone working on the next step to pull the work from the previous step rather than having it pushed on them when the previous step is finished. We will discuss this idea more a little later.

Topics

Kanban method overview

Improving flow

Separate backlog



Continuous flow of work items



Kanban boards focus on improving the flow of issues through the workflow. New issues are continuously added to the backlog and are always being prioritized, usually by the business team. Once the issues are ready to be worked on, they are pushed to the selected for development column and then flow through the statuses of the workflow. Work is continuously being finished before starting new work. When a team member is ready to work on a new issue, they simply choose the top issue under selected for development and drag, or pull, the issue to the in progress column. This ensures that the team is always working on the most important issue as defined by the team.

Improving flow- limit work in progress (WIP)

- **How?**
 - Specify the minimum and/or maximum number of issues allowed in certain project board columns
- **Why?**
 - Better flow
 - Limits waste
 - Promotes teamwork

<https://www.atlassian.com/agile/kanban/wip-limits>



Limiting work in progress is done by specifying the minimum and/or maximum number of issues allowed in certain board columns. Limiting work in progress has many benefits. If you limit work in progress, there are less issues being worked on at one time, which leads to better flow of work getting finished. Because of the work in progress limits, the team focuses on finishing the work in progress before starting new issues. This focus results in less multitasking, which is good because multitasking decreases productivity. The delivery of issues is faster with less work in progress because issues are not piling up in certain columns waiting for work to be restarted. Limiting the amount of work in progress means that issues are sure to achieve the "done" status. This means that the work is really done and can be provided to the customer. Moving issues into the "done" status is the primary measure of progress of the team. By limiting work in progress, any bottlenecks in the process are quickly identified, because there are relatively few issues in progress at any time and problems tend to be easily visible. Identifying and fixing bottlenecks is a way to continuously improve the work of the team.

Limiting work in progress also limits the amount of waste in the process. If there is an inventory buildup of issues in a status, there is waste because work on the issues is delayed. Limiting work in progress also limits the amount of rework that may need to be done if there is a problem caused in an early process step but identified in a later step. There are simply less issues to fix.

Limiting work in progress has the effect of promoting teamwork. Because the number of issues in a status is limited by the WIP limit, the team tends to work together to clear up any blockages.

The link at the bottom of the slide is to a good article on work in progress limits in case you are interested in more discussion on the topic.

Column under minimum limit

The screenshot shows a Jira Kanban board titled "Kanban board" for projectA. The board has four columns: "BACKLOG 2", "SELECTED FOR DEVELOPMENT 1", "IN PROGRESS 0", and "DONE 0". A red arrow points to the "SELECTED FOR DEVELOPMENT 1" column, which is highlighted in yellow. Above the column name, it says "Min 2". The "IN PROGRESS 0" column has "Max 3" written above it. The "BACKLOG 2" column contains two issues: "add feature 2" and "add feature 3", both associated with "PROJ-4". The "SELECTED FOR DEVELOPMENT 1" column contains one issue: "add feature 1", associated with "PROJ-1". The "IN PROGRESS 0" and "DONE 0" columns are currently empty.

Here we have configured a minimum WIP limit. You can see that the selected for development column has a “Min 2” indication next to the column name. Since we only have one issue in this column, it is highlighted in yellow. This is a signal that notifies the team that issues should be added to the column. (Note that minimum WIP limits are currently not supported on Cloud next-gen boards.)

Column over maximum limit

The screenshot shows a Jira Kanban board for projectA / PROJ board. The board has three columns: BACKLOG (0), SELECTED FOR DEVELOPMENT (2 Min 2), and IN PROGRESS (4 Max 3). A red arrow points to the 'Max 3' label above the IN PROGRESS column. The IN PROGRESS column contains four issues: add feature 1, add feature 2, add feature 3, and add feature 4, each associated with a developer icon (PROJ-1, PROJ-4, PROJ-5, PROJ-6).

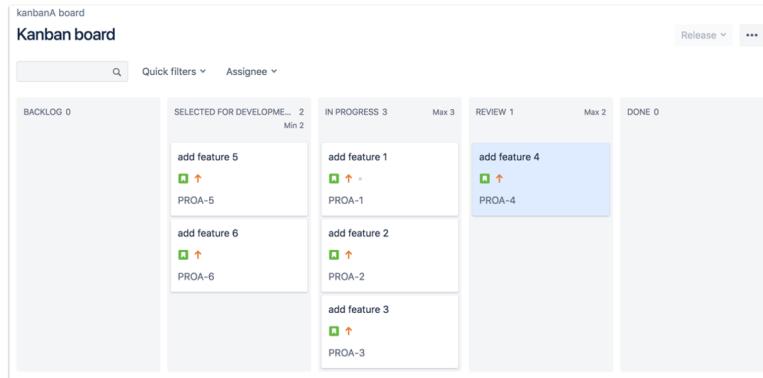
BACKLOG (0)	SELECTED FOR DEVELOPMENT (2 Min 2)	IN PROGRESS (4 Max 3)
	add feature 5 PROJ-7	add feature 1 PROJ-1
	add feature 6 PROJ-8	add feature 2 PROJ-4
		add feature 3 PROJ-5
		add feature 4 PROJ-6

Here we have configured the In Progress column with a maximum WIP limit. You can see the “Max 3” indication next to the column name. We have four issues with a status of In Progress, so the column has turned red. This is a signal to the team to finish the work in progress before starting more issues.

Notice that we don't need to put limits on the backlog or done columns, because those statuses do not contain the team's work in progress.

What should WIP limits be set to?

- Could start with no WIP limits
- Add WIP limits as the process shows problems
- Could set WIP limits to discourage multitasking
- Could set WIP limits on steps that the team neglects



What should work in progress limits be set to? The short answer is that this depends on the specific project and the team. Here are some ideas.

You could start with no work in progress limits and see if the issues are flowing nicely.

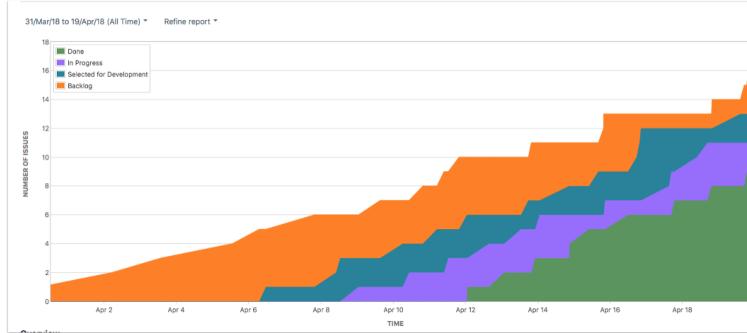
You could add limits as the process starts to show problems. For instance, if sometimes there are no items in the selected for development column, the flow is negatively affected because the team can finish an issue and have no issues to work on. You might want to set a minimum limit on the selected for development column to ensure that this doesn't happen.

You could also set work in progress limits to discourage multitasking. For example, if the team has seven members, you might want to set limits so that each member is working on only one issue at a time.

You could also set work in progress limits on steps that the team tends to neglect. For example, this workflow has a review status after the in progress status. This step ensures that other team members have looked at the work before it is considered done. If you find that issues are piling up in the review column, you could set a maximum limit on that column to ensure that the issues flow all of the way to the done column.

Why reports?

- Visualize the work
- Promote transparency
- Aid troubleshooting and continuous improvement
- Aid planning and estimating

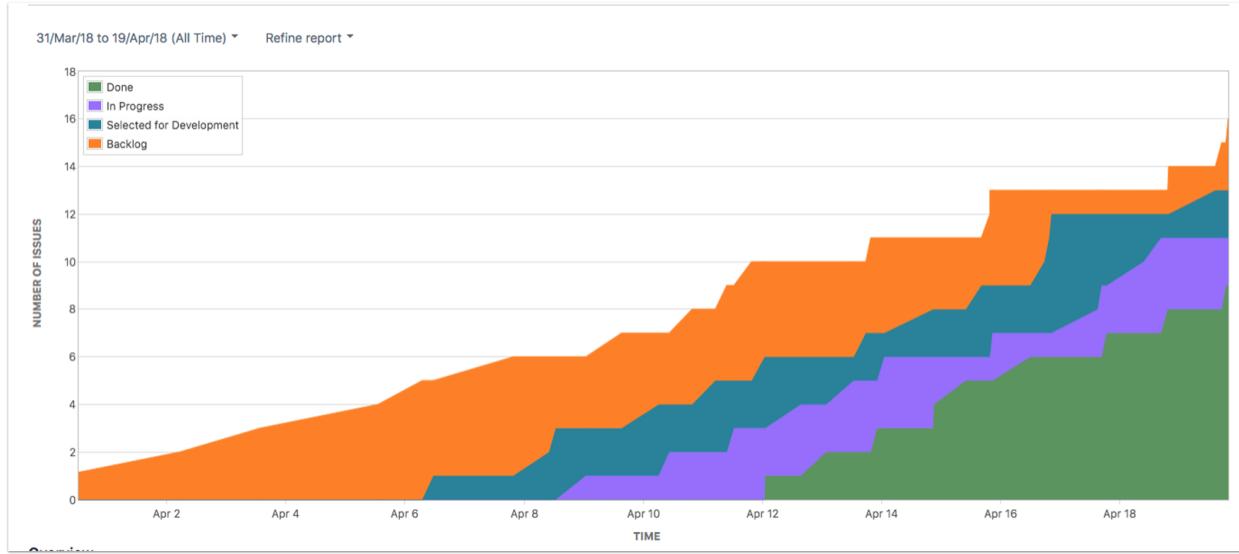


Reports are a key tool related to being agile. Like with boards, agile reports help to visualize the work of the team. This promotes transparency of the project, allowing anyone with access to see and have a common understanding of the current state of the project. The transparency means that there should be no surprises about the delivery of the project.

Reports help with troubleshooting and continuously improving the project. For example, certain reports can clearly show process bottlenecks.

Reports also help with planning the work of the team. They are commonly used in planning meetings to help see the historic progress of the team and provide more accurate planning and estimating.

Cumulative flow diagram



A cumulative flow diagram is a popular report for kanban projects. This is because maintaining and improving the flow of the team is a high priority. Jira provides automatic, real-time reporting. As the team updates issues, the reports are automatically updated. All you have to do is view them.

A cumulative flow diagram shows the number of issues per status over time. You can see from this diagram that for the first few days, all of the issues were in the backlog status, indicated by the orange band. The number of issues in the backlog status increased from 1 to 5 before the first issue was moved to selected for development, which is the second band here. The third band represents issues with the in progress status, so you can see that actual work on an issue started here. The final band indicates issues with a status of done. You can see that the done band increases nicely over time, indicating pretty good flow. You can see that the backlog has almost been depleted, indicated by the shrinking orange band.

You can see many important kanban-related things from a cumulative flow diagram, including bottlenecks in the process, overall flow improvement and how well the team is keeping up with the backlog. You can't really see individual issue problems with this diagram- you can use other reports or views to focus on specific issues.

Topics

Kanban method overview

Improving flow

Separate backlog



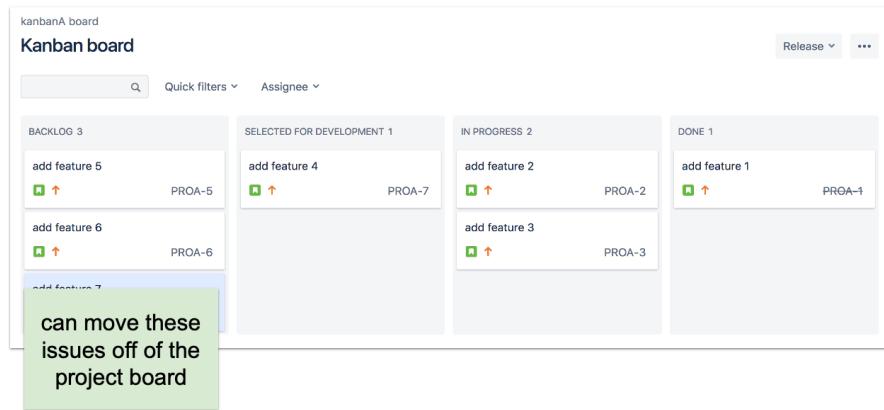
Separate backlog

The screenshot shows a Jira Kanban board titled "kanbanA board" with the section "Kanban board". The board has four columns: "BACKLOG 3", "SELECTED FOR DEVELOPMENT 1", "IN PROGRESS 2", and "DONE 1". The "BACKLOG 3" column contains three items: "add feature 5" (PROA-5), "add feature 6" (PROA-6), and "add feature 7". A green callout box is overlaid on this column with the text "can move these issues off of the project board". The other three columns each have one item: "add feature 4" (PROA-7) in "SELECTED FOR DEVELOPMENT 1", "add feature 2" (PROA-2) in "IN PROGRESS 2", and "add feature 1" (PROA-1) in "DONE 1". The top right of the board has buttons for "Release" and "...".

In Jira, the backlog column on a kanban board can be separated from the rest of the board. This has a few advantages. It allows the development team to only see and focus on issues that they can work on, because issues in the backlog column are not ready to be worked on. The backlog column also can become quite long and hard to manage in this format. A separate backlog column is easier to manage, and any work there is not visible to the rest of the team.

Why a separate backlog?

- Simplicity- separates the planning of issues from the project board
- The team can focus on work items that are ready to be worked on



Separating the backlog from the project board separates the planning of the backlog from the work of the team on the project board. Since the team only can work on issues once they are in the selected for development column, the backlog can be removed so that the board only shows issues that can be worked on.

Managing the separate backlog

The screenshot shows the Jira interface for a project named 'projectA'. The sidebar on the left has a red arrow pointing to the 'Backlog' tab. The main area displays a 'Selected for Development' section with one issue: 'add feature 4' (PROA-7). Below it is a 'Backlog' section with three issues: 'add feature 5' (PROA-5), 'add feature 6' (PROA-6), and 'add feature 7' (PROA-8). A 'Create issue' button is located at the bottom of the backlog list.



When you click on the Backlog tab, you see the backlog. These are all of the issues with the status that was dragged to the kanban backlog when configuring the board. In this case, these are the issues with a status of backlog. Notice that the items in the backlog are in a list and you have a lot of room to edit the backlog.

Above the backlog is the first column of the board. This is there so that you can easily move issues from the backlog onto the board, allowing the team to see the issues. Also, if you would edit an issue here, changing the status to any column on the kanban board, the issue will move out of the backlog and onto the kanban board.

Takeaways



- A project board should have a continuous flow of issues moving from backlog to done columns
- Work in progress limits can improve the flow of value by focusing the team
- In Jira, the backlog can be separated from the project board, simplifying the board and allowing separate backlog work



Tasks

Kanban Projects



- Configure WIP limits
- View a cumulative flow diagram
- Configure a separate backlog



6

Scrum Projects



What will you learn?

- Identify scrum artifacts
- Create and execute sprints
- View scrum reports



Topics

Scrum overview

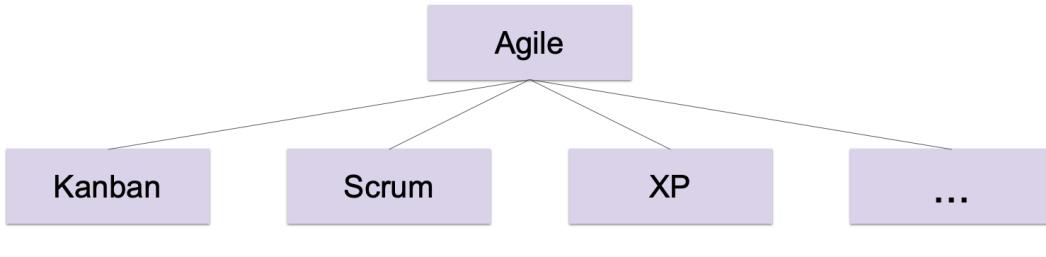
Creating and executing sprints

Scrum reports



What is scrum?

- "Scrum is a framework for developing, delivering, and sustaining complex products." Scrum Guide 2017
 - <https://www.scrum.org/resources/scrum-guide>
- A way of achieving agility

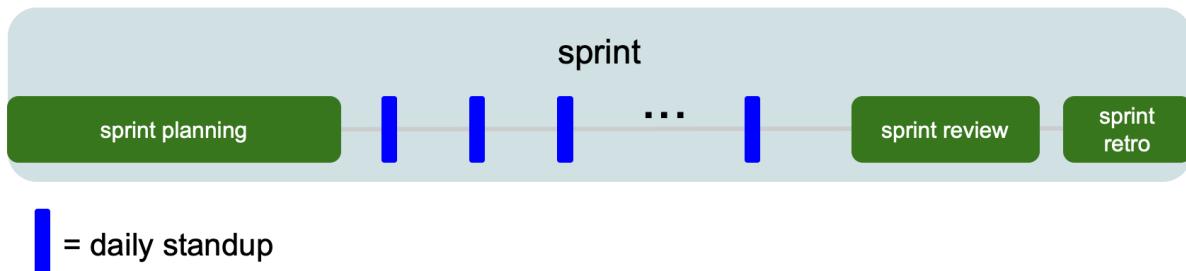


According to the Scrum Guide, which is written by the creators of scrum, scrum is a framework for developing, delivering, and sustaining complex products. It is a relatively simple framework for dealing with complex, unpredictable projects. By framework, we mean that scrum contains basic structure and ideas for completing a project. The Scrum Guide refers to scrum as a process framework for your project management and work techniques, rather than a standalone process or definitive method. Every team and every project is different, so the basic ideas of scrum can be customized to suit your specific project. This is contrasted to a rigid methodology in which every project is executed the same way.

Much of what we will discuss here is discussed in the The Scrum Guide. It is free, relatively short and very well written. It is highly recommended that you read it. Scrum is a way of achieving the idea of agility.

Previously, we have seen that kanban is another option. You can think of agile as a mindset, and the methods such as kanban, scrum and XP as ways of achieving agility.

Scrum events



A sprint contains four types of events. You may also hear events referred to as ceremonies or simply meetings. The events are the sprint planning meeting, the daily standups, the sprint review and the sprint retrospective. Scrum events occur at regular intervals and minimize the need for other meetings. These events are designed to maximize the opportunities for feedback and continuous learning, and are a key part of achieving agility. In between these meetings is where the work of the issues of the sprint is completed.

We will discuss each of these meetings separately.

<https://www.atlassian.com/agile/scrum/ceremonies>

Parts of the scrum framework

- **Artifacts**- product backlog, sprint backlog, sprint goal, sprint board, reports
- **Roles**- product owner, scrum master, development team members, stakeholders
- **Events/Meetings/Ceremonies**- sprint, sprint planning meeting, daily standups, sprint review, sprint retrospective



There are three main parts of the scrum framework.

Artifacts are tools that allow for transparency of the project. They allow anyone with access to them to see the current state of the project. The artifacts that we will talk about in this module are the product backlog, the sprint backlog, the sprint goal, the sprint board and sprint reports.

The second part of the scrum framework is the roles related to scrum. In the next module, we will discuss the roles of product owner, scrum master, development team members and stakeholders.

The third main part of the scrum framework is the events related to scrum. These are also called ceremonies or meetings. The sprint guide considers the sprint as a container event for the other events. In the next module, we will discuss the sprint planning meeting, daily standups, the sprint review and the sprint retrospective.

Topics

Scrum overview

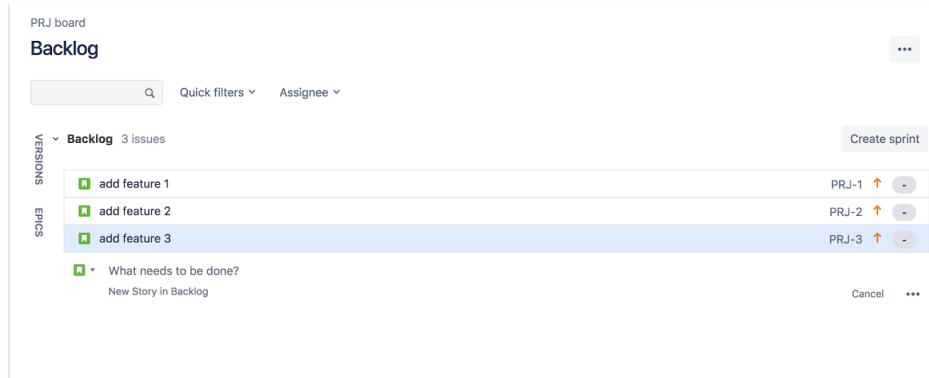
Creating and executing sprints

Scrum reports



Product backlog

- An ordered, ever-changing to do list for the project
- Can include features, improvements, bug fixes, etc.
- Issues near the top should include more detail



A product backlog is an ordered, ever-changing to do list for the project. It contains issues that are not yet part of any sprint. The Scrum Guide refers to issues as items. You might also hear them referred to as stories. Constant feedback means that the product backlog is always changing.

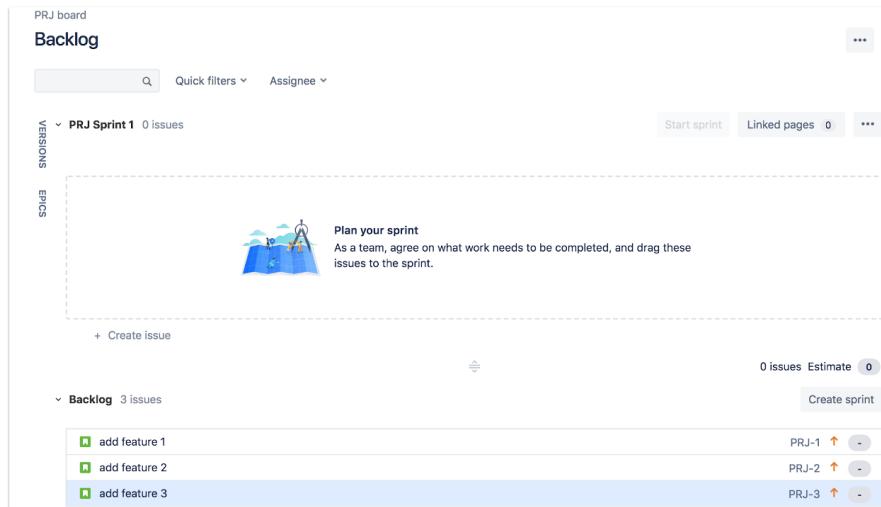
Here you can see the product backlog in Jira. It is located under the Backlog tab for scrum projects. This product backlog contains three issues. When you first add issues to a scrum project, they are automatically placed in the product backlog.

The product backlog can include issues that represent features, improvements, bug fixes and any other type of issue that you would like.

The product backlog is ordered. Issues near the top of the backlog are the closest to being worked on, so they usually have more details than the lower items.

Modifying the product backlog is called product backlog refinement. You might also hear this referred to as backlog grooming. According to the Scrum Guide, each scrum team decides how to do product refinement, but it should consume no more than 10 percent of the development team's time.

Creating a sprint with Jira



To create a sprint in Jira, you click the Create sprint button in the product backlog. After clicking the Create sprint button, Jira creates an empty sprint. You can see here that Jira named the sprint using the project key, which is PRJ in this example, and the sprint number, which is one in our case. You can see that Jira invites you to drag issues from your product backlog into the sprint.

Sprint backlog

- A subset of the product backlog
- The list of issues to be completed in the sprint
- Includes the plan on how to accomplish the work of the issues

The screenshot shows the Jira 'Backlog' screen for a project named 'PRJ'. At the top, there are filters for 'Quick filters' and 'Assignee'. Below that, a section titled 'VERSIONS' shows 'PRJ Sprint 1' with '2 issues'. A red box highlights two issues: 'add feature 2' (PRJ-2) and 'add feature 1' (PRJ-1). To the left of this section, the text 'sprint backlog' is overlaid in red. Below this, under 'What needs to be done?', there is a 'New Story in PRJ Sprint 1' button. Further down, another section titled 'Backlog' shows '1 issue' with 'add feature 3' (PRJ-3). At the bottom right, there is a 'Create sprint' button.

Here, we have dragged two of the issues from the product backlog to the sprint. The list of issues to be completed during the sprint is called the sprint backlog. The sprint backlog includes a plan on how to accomplish the work of the issues. In Jira, this means that before starting the actual sprint, more details are added to the issues in the sprint backlog. Those details describe how the work of the issues will be done. We will discuss more about adding these details later in the course

Estimation- story points

- Story points are a relative measure of the amount of work required to complete the story
- Used to help decide how many stories can be completed in the sprint

The screenshot shows a Jira Backlog board titled 'PRJ board' with the section 'Backlog'. At the top, there are filters for 'Quick filters' and 'Assignee'. Below this, the 'PRJ Sprint 1' backlog is shown with 2 issues: 'add feature 2' (PRJ-2, 1 point) and 'add feature 1' (PRJ-1, 2 points). A total estimate of 3 points is displayed. To the right of the backlog, a detailed view of an issue titled 'add feature 1' is open. This view includes fields for 'Status' (To Do), 'Assignee' (Unassigned), 'Labels' (None), and 'Story points' (2). There are also buttons for 'Start sprint', 'Linked pages', and three dots for more options.

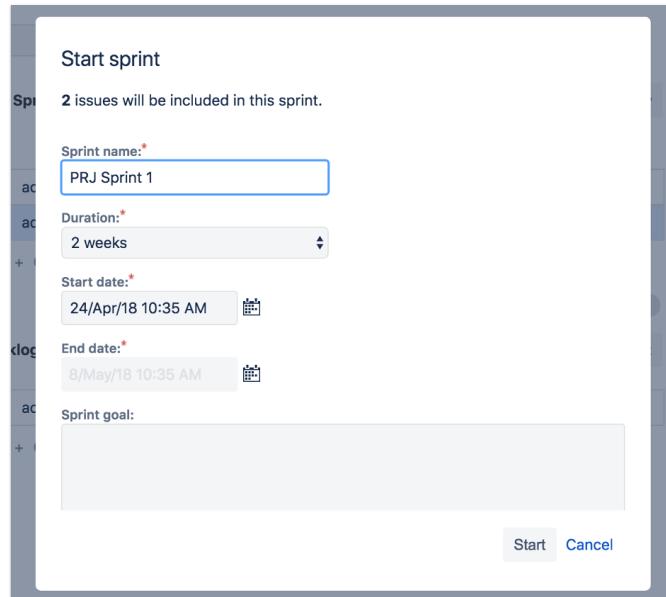
As part of planning for sprints, it is common to estimate how much work an issue will take. Story points are the most common estimation statistic. In Jira, you can use story points, hours, issue count or create your own estimation statistic.

Story points are a relative measure of the amount of work required to complete an issue. For example, an issue that is assigned two story points is assumed to take about twice as long to complete as an issue that is assigned one story point.

You can see that in Jira, there is a field on each issue named story points. Here we have assigned 2 story points to this issue. In the sprint backlog, you can see that the story points are shown in the gray boxes, along with a total estimate of 3 points for the sprint backlog. You can see that the issue remaining in the product backlog has been assigned four story points.

Story points are used to help the team decide how many issues can be completed in a sprint.

Sprint details



The start sprint screen appears. Jira starts by reminding you that you have added two issues to the sprint backlog. You can modify the sprint name, specify the sprint's duration and specify the start date for the sprint. You can see that you can have the sprint started at a later date, so you don't have to actually click the start sprint button on the first morning of the sprint. You could also set up multiple sprints at one time.

Sprint goal

- Represents the objective of the sprint's increment
- Is reached by completing the sprint backlog
- Does not change during the sprint
- The sprint is a success if the sprint goal is reached

Start sprint

2 issues will be included in this sprint.

Sprint name: * PRJ Sprint 1

Duration: * 2 weeks

Start date: * 24/Apr/18 10:35 AM

End date: * 8/May/18 10:35 AM

Sprint goal: Create the first product features



The start sprint window also contains a place to enter what is called the sprint goal. The sprint goal represents the objective of the sprint's increment. It is agreed to by the team.

The sprint goal is reached by completing the issues in the sprint backlog.

A scrum rule is that the sprint goal does not change during the sprint.

The sprint is considered a success if the sprint goal is reached.

Sprint board

Only contains issues from the sprint backlog

The screenshot shows a Jira Sprint Board titled "PRJ Sprint 1". The board is set up with three columns: "TO DO", "IN PROGRESS", and "DONE". In the "TO DO" column, there are two items: "add feature 2" and "add feature 1". Both items have a green checkmark icon and a small orange arrow icon. The first item is assigned to "PRJ-2" and has a value of 1. The second item is assigned to "PRJ-1" and has a value of 2. The board also includes a sidebar with project navigation options like Backlog, Active sprints, Reports, Releases, Issues, Pages, Components, Add item, and Settings.

A sprint has a sprint board. Notice that it only contains the issues in the sprint backlog. Issues in the product backlog, or issues that are assigned to other sprints are not shown on the sprint board. Even in sprint projects, boards are often called kanban boards, so don't be confused if you hear that term related to a sprint.

Topics

[Scrum overview](#)

[Creating and executing sprints](#)

[**Scrum reports**](#)



Scrum reports- burndown chart



<https://www.atlassian.com/agile/tutorials/burndown-charts>



A burndown chart shows the progress that the team makes during a sprint. The sprint backlog starts with a certain number issues, each with an associated number of story points or other estimation statistic. The total number of starting story points is shown on the left of the chart. This is the number of story points that the development team estimated that it would complete in the sprint. In our case, this sprint has 3 story points. The gray guideline shown is used to show the number of story points that should remain on a given day, assuming a linear burndown of story points. On the last day of the sprint, the guideline reaches zero story points. This means that the work of the sprint should be finished on the last day. Notice that the non-working days are shown in the chart, and the guidelines assume no progress will be made on those days.

As the sprint is underway, Jira will automatically update the burndown chart as the status of the issues is updated by the team. The red line shows the actual number of remaining story points over time. You can see that about two days into this sprint, one story point was completed. On the last day of the sprint, the remaining two story points were completed. Consulting this chart is an easy way to see if the team is on track for the current sprint. If the red line is below the gray line, your team is on track to complete all of the story points and reach the sprint goal. If not, the team may need to make some adjustments to still reach the sprint goal.

The link shown contains more information on burndown charts.

Scrum reports- sprint report

PRJ board / Reports

Sprint Report

PRJ Sprint 1

Active sprint 24/Apr/18 11:13 AM - 08/May/18 11:13 AM [Linked pages](#)

Create the first product features.

The burndown chart shows the progress of the sprint. The Y-axis represents story points from 0 to 3.5. The X-axis shows dates from April 24 to May 8. A line graph starts at 3.0 on April 24, drops to 2.0 by April 26, remains flat until April 28, then drops to 1.0 by April 30. It stays at 1.0 until May 2, then drops to 0.5 by May 4. It remains at 0.5 until May 6, then drops to 0.0 by May 8. There are two vertical grey bars representing the sprint duration from April 24 to May 8.

Status Report

[View in Issue Navigator](#)

Issues Not Completed

Key	Summary	Issue Type	Priority	Status	Story Points (3)
PRJ-1	add feature 1	Story	Medium	To Do	2
PRJ-2	add feature 2	Story	Medium	To Do	1



The sprint report contains a nice summary of the sprint. It shows the burndown chart, as well as the current status of all of the issues in the sprint. This is an easy way to see how the sprint is progressing.

Velocity

Represents the rate at which the team accomplishes work

- Usually it is the number of story points completed per sprint
- In this example, the velocity is 3 story points per sprint



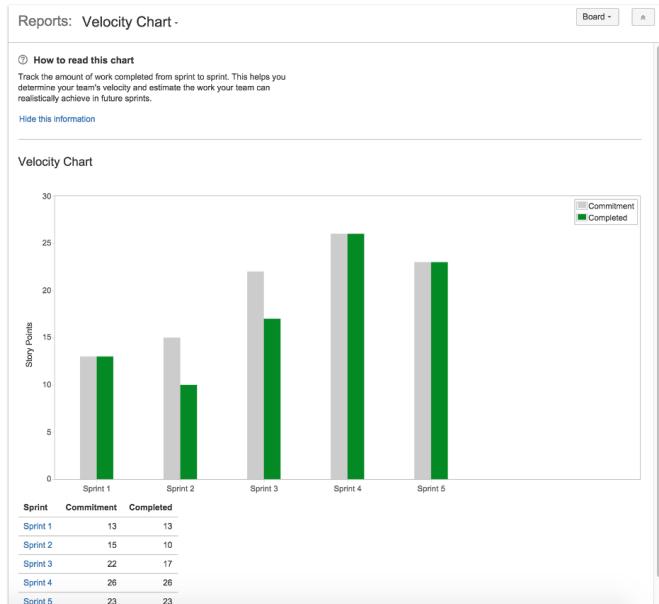
Velocity represents the rate at which the team accomplishes work.

Usually it is the number of story points completed per sprint. Some teams use an estimation statistic other than story points, so in that case velocity measures some other units completed per sprint.

You can see the team's velocity of a single sprint by looking at a burndown chart. In this sprint, the team completed three story points in the sprint, so its velocity is 3.

Scrum reports- velocity chart

Shows the estimated and actual velocity of the team over time



The velocity chart shows the estimated and actual velocity of the team over time. In this example, prior to the first sprint, the team estimated that it would complete 13 story points, shown with the gray bar. It actually completed those 13 story points, as shown by the green bar. The velocity of the team for sprint one was 13 story points. In sprint 2, the team optimistically estimated that it could complete 15 story points. It actually completed 10.

The velocity chart shows a quick snapshot of the change in the team's productivity over time. As the team improves, its velocity generally improves, so velocity charts usually trend upward.



Takeaways

- Scrum is an agile framework
- Jira includes scrum artifacts such as product backlogs, sprint backlogs and sprint boards
- Common sprint reports include burndown charts and velocity charts



Tasks

Scrum

- Create a scrum project
- Create issues in the product backlog
- Create and plan a sprint
- Execute a sprint
- Complete a sprint



7

Quick Search and Basic Search



What will you learn?

- Identify the ways to search in Jira
- Use quick search
- Use basic search



Topics

Searching overview

[Quick search](#)

[Basic search](#)



Viewing a project's progress



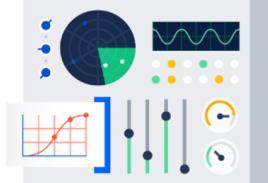
Boards



Search



Reports



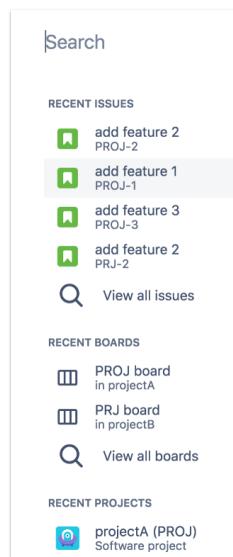
Dashboards



Project boards, searching, reports and dashboards help your team view the current and historic status of the project, as well as help plan the project. Using these tools, you can answer most questions that are relevant to the team, either in an ongoing manner or with specific just-in-time team questions. We will cover searching now, but the same techniques are used behind the scenes for reports and dashboards.

Searching

1. Quick search



There are several topics related to searching in Jira. We will discuss them quickly now, and then discuss them in more detail in this and the next module. The first is called a quick search. This allows you to quickly search for text in issues, board names, project names and filter names using a text-based search. You can also quickly access recently used items, as shown here.

Searching

1. Quick search
2. Basic search (user interface elements)

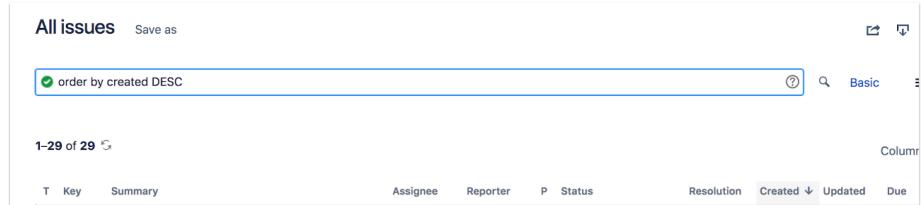
The screenshot shows the Jira 'All issues' search results page. At the top, there are search filters: 'Project: All', 'Type: All', 'Status: All', 'Assignee: All', 'Contains text' (with a dropdown arrow), 'More' (with a dropdown arrow), a magnifying glass icon, and an 'Advanced' link. Below the filters, it says '1-29 of 29' with a refresh icon. At the bottom, there is a table header with columns: T, Key, Summary, Assignee, Reporter, P, and Status.



Another type of search is called basic search. In a basic search, issues are searched for using a row of user-friendly interface elements, as shown here. Changes to these elements will change the resulting list of issues.

Searching

1. Quick search
2. Basic search (user interface elements)
3. Advanced search (JQL)



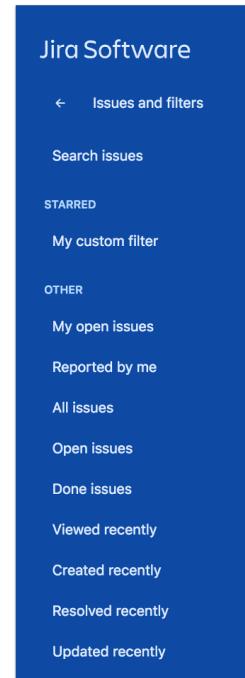
The screenshot shows the Jira search interface for advanced search (JQL). At the top, there is a search bar with the placeholder "order by created DESC". Below the search bar, the results are displayed with the message "1-29 of 29". The results table has columns for T (Ticket), Key, Summary, Assignee, Reporter, P (Priority), Status, Resolution, Created (sorted descending), Updated, and Due. The "Created" column is currently sorted in descending order.



Advanced search is an alternative to basic search, where you can specify your search using Jira Query Language, or JQL. This is a way to execute more complicated searches than with basic search.

Searching

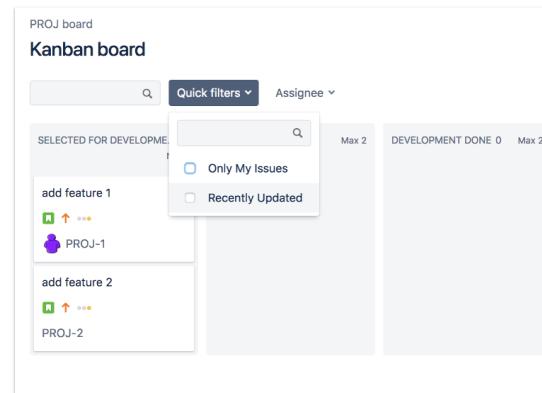
1. Quick search
2. Basic search (user interface elements)
3. Advanced search (JQL)
4. Filters



Filters are used to limit the issues that are displayed in search results. These are the default filters provided by Jira. For example, if you click on the Open issues filter, the results will show all issues in all projects that have not been closed. You can also add custom filters, such as the filter called My custom filter here. Even though filters are usually presented as user interface elements, they have an underlying JQL statement associated with them. That JQL statement defines which issues to show.

Types of searching

1. Quick search
2. Basic search (user interface elements)
3. Advanced search (JQL)
4. Filters
5. Quick filters



The final search topic that we will discuss is quick filters. Quick filters are used to limit which issues are shown on a board. Here are the two out of the box quick filters, used to show only your issues or recently updated issues. You can also add custom quick filters to the board. Like standard filters, quick filters have an underlying JQL statement associated with them.

Why is searching important?

Adapt your Jira experience to your team's processes

The screenshot shows a Jira interface for a project named 'projectA'. On the left, there's a sidebar with various project management options like 'PROJ board', 'Backlog', 'Kanban board', 'Reports', etc. The 'Kanban board' option is currently selected. The main area displays a 'PROJ board' with a 'Kanban board' view. There are two columns: 'SELECTED FOR DEVELOPMENT' and 'DEVELOPMENT DONE'. Under 'SELECTED FOR DEVELOPMENT', there are two items: 'add feature 1' and 'add feature 2', each associated with a small icon and the identifier 'PROJ-1' and 'PROJ-2' respectively. A modal window titled 'Quick filters' is overlaid on the board, containing two checkboxes: 'Only My Issues' and 'Recently Updated', neither of which is checked. The overall theme is light blue and white.



We are discussing searching, JQL and filters because they are used to help adapt your Jira experience to your team's processes. We have seen that each agile team is unique and constantly improving, so the tools that are used need to be flexible enough to adapt to changing circumstances.

Topics

Searching overview

[Quick search](#)

Basic search



Quick search

The screenshot shows two side-by-side views of the Jira quick search interface: Cloud on the left and Server on the right.

Cloud View:

- A large search input field at the top labeled "Search".
- A "RECENT ISSUES" section containing six items, each with a green icon:
 - add feature 1 PROJ-1
 - add feature 2 PRJ-2
 - add feature 1 PRJ-1
 - add feature 3 PRJ-3
 - add feature 2 PROJ-2
 - add feature 3 PROJ-3
- A "View all issues" button with a magnifying glass icon.

Server View:

- A header bar with "Search" and other navigation icons.
- A "ISSUES" section listing recent issues:
 - add item 5a PROJ-7
 - add item 2 PROJ-2
 - add item 2 PROJF-2
 - add item 1 PROJF-1
 - initial release PROJF-4
 - add item 3 PROJF-3
- A "View all issues" button.
- A "PROJECTS" section listing recent projects:
 - projectA (PROJ) Software
 - projectfinal (PROJF) Software
- A "View all projects" button.

Both views have a "Cloud" and "Server" label below them, and a small blue triangle icon in the bottom right corner.

When you initially open the quick search, the results show recent items. You can click on an item to display it.

Quick search does not return just issues. When you open quick search, you can also access recent boards, projects and filters. When you perform a search, all of these items that match will be displayed.

Quick search- with search terms and keywords

The image displays three separate Jira search results boxes, each showing a list of issues containing specific search terms:

- feature**: Shows 6 issues. Issues include: add feature 2 PROJ-2, add feature 3 PROJ-3, add feature 1 PROJ-1, add feature 2 PRJ-2, add feature 3 PRJ-3, and add feature 1 PRJ-1.
- feature NOT 1**: Shows 4 issues. Issues include: add feature 2 PROJ-2, add feature 3 PROJ-3, add feature 2 PRJ-2, and add feature 3 PRJ-3.
- feature OR sample**: Shows 6 issues. Issues include: add feature 3 PROJ-3, add feature 2 PROJ-2, add feature 3 PRJ-3, add feature 2 PRJ-2, Instructions for deleting this sample SAM-17, and add feature 1 PROJ-1.

<https://confluence.atlassian.com/jiracorecloud/search-syntax-for-text-fields-765593720.html>



Usually you will just be entering simple text when searching with quick search. You also have the option of adding slightly more sophisticated syntax. As an example, on the left, we are searching for the text "feature". You can see that this text is in the summary for six issues in our Jira account.

We can exclude feature one by adding a capitalized NOT followed by the number one. You can see that the two issue containing feature one are excluded from the results. In the search, NOT must be all caps. The search terms are not case sensitive, but text-field search keywords such as NOT, OR and AND must be capitalized.

In this example, we use the capital OR keyword to search for "feature" or "sample".

For more information on this type of searching, see the link shown here on search syntax for text fields.

Topics

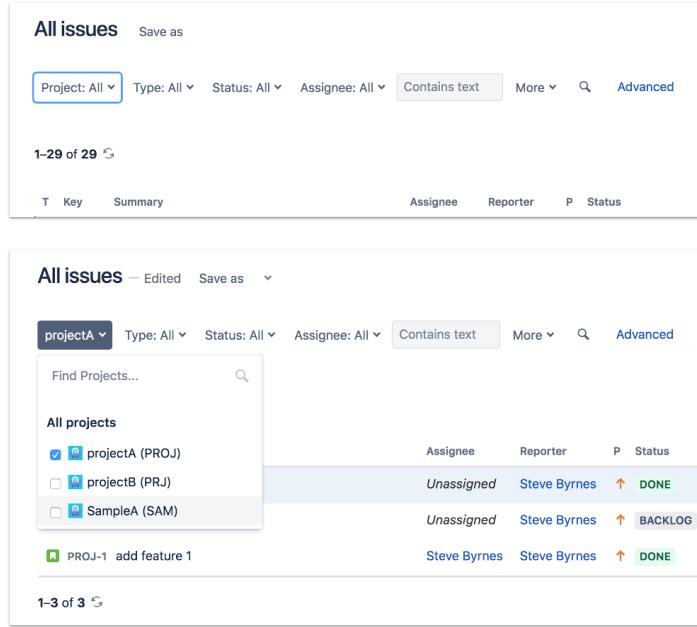
Searching overview

Quick search

Basic search



Basic search



The screenshot shows the Jira issue navigator with the title "All issues". At the top, there is a search bar with dropdowns for "Project: All", "Type: All", "Status: All", "Assignee: All", "Contains text", "More", and an "Advanced" link. Below the search bar, it says "1-29 of 29" with a refresh icon. The main area has columns for "T", "Key", "Summary", "Assignee", "Reporter", "P", and "Status". A table below lists three issues:

T	Key	Summary	Assignee	Reporter	P	Status
	Unassigned	Steve Byrnes	Steve Byrnes	Steve Byrnes	↑	DONE
	Unassigned	Steve Byrnes	Steve Byrnes	Steve Byrnes	↑	BACKLOG
	PROJ-1	add feature 1	Steve Byrnes	Steve Byrnes	↑	DONE

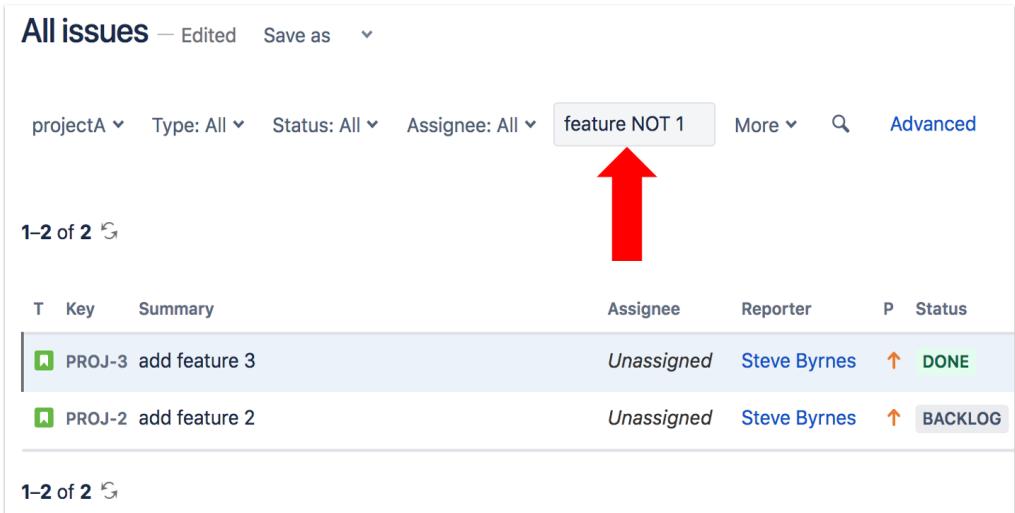
At the bottom left, it says "1-3 of 3" with a refresh icon. On the right side of the interface, there is a blue triangle icon.

Basic search uses user-friendly interface elements in the issue navigator to search for issues. This row of elements shows that we are in the basic search. We could click on the Advanced link to switch to the advanced search. Likewise, if we are in the advanced search, we can click on the Basic link to switch to basic search.

You can see that this search searches all projects for issues of any type in any status and with any assignee. This is searching for all issues in the Jira account. You can see that there are currently 29 issues in this account, assuming that we have permission to see all issues.

If we click on the project dropdown and choose project A, you can see that three issues are returned.

Contains text



The screenshot shows the Jira 'All issues' search results page. At the top, there are dropdown filters for 'projectA', 'Type: All', 'Status: All', 'Assignee: All', and a search bar containing the text 'feature NOT 1'. A red arrow points upwards from the bottom of the page towards this search bar. Below the filters, the text '1-2 of 2' is displayed. The main area shows a table with two rows of issue data. The columns are labeled 'T', 'Key', 'Summary', 'Assignee', 'Reporter', 'P', and 'Status'. The first row contains issue 'PROJ-3 add feature 3' with status 'DONE'. The second row contains issue 'PROJ-2 add feature 2' with status 'BACKLOG'. Both issues are unassigned and reported by Steve Byrnes.

T	Key	Summary	Assignee	Reporter	P	Status
	PROJ-3	add feature 3	Unassigned	Steve Byrnes	↑	DONE
	PROJ-2	add feature 2	Unassigned	Steve Byrnes	↑	BACKLOG

Notice that basic search includes a text box. This behaves much like the quick search, but is limited to searching issues. If you enter text in this textbox, Jira will search in fields that typically hold text, such as summary or description. In this example, we have changed the project dropdown to project A and added "feature NOT 1" to the "contains text" box. This will filter the "add feature 1" issue from the project, and you can see that we are left with two issues. You can see that NOT in the quick search can also be used here.

Searching more fields

The screenshot shows the Jira 'All issues' search results page. At the top, there are filters for Project, Type, Status, Assignee, and a 'Contains text' search bar. Below the filters, it says '1-29 of 29'. The main area displays a table with columns for T, Key, Summary, and Assignee. The first row has 'PRJ-3' as the key and 'add feature 3' as the summary, with 'Unassigned' under Assignee. The second row has 'PRJ-2' and 'add feature 2', with 'Steve Byrnes' assigned. The third row has 'PRJ-1' and 'add feature 1', also with 'Steve Byrnes' assigned. The fourth row has 'PROJ-3' and 'add feature 3', with 'Unassigned' again. To the right of the table, a 'More' dropdown menu is open, showing a list of recent criteria: 'Summary', 'Organizations', 'Description', 'Resolution', 'Updated Date', and 'Created Date'. There is also a link to 'All Criteria' and a note about excluding 7 hidden items.

If you want to search for values of fields that are not listed in basic search, you can click on the More dropdown to add searches for other fields. We will click on the updated date checkbox to add a search related to the day that an issue was modified.

Specifying a priority

The screenshot shows a Jira search results page with three items listed: PROJ-3 add feature 3, PROJ-2 add feature 2, and PROJ-1 add feature 1. A dropdown menu titled 'Priority: All' is open over the first item. The menu includes a search bar labeled 'Find Priorities...' and five priority levels: Highest (red up arrow), High (orange up arrow), Medium (orange up arrow), Low (green down arrow), and Lowest (green down arrow). The 'Medium' option is currently selected.

T	Key	Summary	Reporter	P	Status	R
	PROJ-3	add feature 3	Steve Byrnes	↑	DONE	D
	PROJ-2	add feature 2	Steve Byrnes	↑	BACKLOG	L
	PROJ-1	add feature 1	Steve Byrnes	↑	DONE	D

Here we have selected priority from the More dropdown menu. Notice that Jira will automatically provide the appropriate interface depending on the type of field chosen. We have just seen that when we selected the updated date field, we were provided an interface related to dates. In this case, we have selected the Priority field, so Jira provides us a set of priorities to choose from.

Specifying an updated date

The screenshot shows the Jira search interface with various filters applied: Project: All, Type: All, Status: All, Assignee: All, Contains text (with the search term 'PRJ-1'), More, Advanced, and Updated Date: All. The Updated Date dialog is open, showing the 'Within the last 3 days' option selected. Other options include 'More than [] minutes ago', 'Between [] and []', and 'In range [] to []'. Below the dialog are 'Update' and 'Close' buttons. To the right, a search results table displays three issues:

Assignee	Reporter	P	Status
Unassigned	Steve Byrnes	↑	IN PROG
Steve Byrnes	Steve Byrnes	↑	DONE
Steve	Steve	↑	DONE

Here we have clicked on Updated Date, and you can see that Jira adds a dialog allowing you to specify the date values to search for. We have chosen to search for any issues that have been updated within the last three days.



Takeaways

- Quick search can search the text of issues, board names, project names and filter names
- Basic search is a user-friendly way to search for issues



Tasks

Quick Search and Basic Search

- Perform quick searches
- Perform basic searches



8

JQL



What will you learn?



- Describe Jira Query Language (JQL)
- Write JQL using autocomplete
- Use functions in JQL queries



Topics

[JQL overview](#)

[Autocomplete](#)

[Functions](#)



Basic and advanced/JQL search

basic

The screenshot shows the Jira basic search interface. At the top, there's a search bar with the word "basic". Below it, a search panel includes dropdowns for "Project: All", "Type: All", "Status: All", "Assignee: All", and a "Contains text" field. There are also "More" and "Advanced" links. The results section shows "1-29 of 29" issues. A table header row includes columns for "T", "Key", "Summary", "Assignee", "Reporter", "P", and "Status".

advanced/JQL

The screenshot shows the Jira advanced search interface. The search bar contains the query "order by created DESC". The results section shows "1-29 of 29" issues. A table header row includes columns for "T", "Key", "Summary", "Assignee", "Reporter", "P", "Status", "Resolution", "Created", "Updated", and "Due".



Previously, we explored basic search. In this example, we are searching for all issues that we have permission to view in the Jira account, because none of the search elements are being selective. On a basic search, there is an Advanced link, which allows you to switch to the advanced mode of searching for issues.

The bottom image shows advanced search. Jira will automatically populate the textbox with a text-based equivalent for the current basic search. This text says "order by created DESC". In other words, search for all issues in this Jira account and sort the resulting list by the value of the created field. The created field holds the time and date that the issue was created. Descending means to show the newest issue at the top of the resulting list and the oldest issue at the bottom. You can click the Basic link to switch back to the basic search.

JQL



Jira Query Language (JQL)- Searches issues and orders results only



That text that we saw on the advanced issue search is called Jira Query Language, or JQL. JQL is used to search issues and order the results only.

Basic vs. advanced/JQL search



Basic search

- User-friendly interface
- Queries can be complex, but there are limits



Advanced/JQL search

- Uses JQL
- Most powerful search method
- JQL can be used in automation scripts



We have seen that the basic search contains user-friendly interface elements to perform the search. We have also seen that the queries can be quite complex by selecting from multiple elements or adding text-based searching.

Advanced search uses JQL and allows for more powerful searches because it is not limited by the capabilities of the user interface elements of basic search. Also, if you are automating anything related to Jira, you would use JQL, because automation relies on text-based scripts.

Whether you are using basic or advanced search, the JQL related to the search is always there behind the scenes.

"Writing" JQL- the easiest way

The screenshot shows two side-by-side Jira search interfaces. On the left, under 'Basic search', the search bar contains 'projectA' and the results show three issues: PROJ-3, PROJ-2, and PROJ-1, all assigned to Steve Byrnes and reported by Steve Byrnes, with statuses DONE, BACKLOG, and DONE respectively. On the right, under 'Advanced search', the search bar contains the JQL query 'project = PROJ order by created DESC'. The results are identical to the basic search, showing the same three issues in the same order.

T	Key	Summary	Assignee	Reporter	P	Status
	PROJ-3	add feature 3	Steve Byrnes	Steve Byrnes	↑	DONE
	PROJ-2	add feature 2	Steve Byrnes	Steve Byrnes	↑	BACKLOG
	PROJ-1	add feature 1	Steve Byrnes	Steve Byrnes	↑	DONE

T	Key	Summary
	PROJ-3	add feature 3
	PROJ-2	add feature 2
	PROJ-1	add feature 1

The easiest way to write a JQL query is to not write it at all. You can let Jira write it for you. Here we are in basic search and select project A from the dropdown. We can see that project A has 3 issues. We can then click on the Advanced link to enter advanced search.

The search results don't change at all, but a query is displayed instead of the basic search user interface. That query displays all of the issues with a project key of PROJ. The results are ordered by the date that the issue was created.

The two main parts of a JQL query



project = PROJ order by created DESC



*search clause- selects a
subset of issues*

*order by clause- orders the
results*



JQL queries can fundamentally do two things, and we see both of them in the query that was just created. The search clause can select a subset of issues, as we see in the first part of the query. We are selecting only issues in which the project field has a value of PROJ. And they can sort the results using an “order by” clause. We are ordering the results by the date that they were created.

The simplest JQL query

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
	SAM-23	As a user, I'd like a historical story to show in reports	Steve Byrnes	Steve Byrnes	↑	DONE	Done	27/May/18	09/Jun/18	...
	SAM-22	As a user, I'd like a historical story to show in reports	Steve Byrnes	Steve Byrnes	↑	DONE	Done	27/May/18	07/Jun/18	
	SAM-21	As a user, I'd like a historical story to show in reports	Steve Byrnes	Steve Byrnes	↑	DONE	Done	27/May/18	04/Jun/18	
	SAM-20	As a user, I'd like a historical story to show in reports	Steve Byrnes	Steve Byrnes	↑	DONE	Done	27/May/18	02/Jun/18	
	SAM-19	As a user, I'd like a historical story to show in reports	Steve Byrnes	Steve Byrnes	↑	DONE	Done	27/May/18	31/May/18	
	SAM-18	As a user, I'd like a historical story to show in reports	Steve Byrnes	Steve Byrnes	↑	DONE	Done	27/May/18	28/May/18	
	SAM-17	Instructions for deleting this sample board and project are in the description for this issue >> Click the "SAM-17" link and read the description tab of the detail view for more	Steve Byrnes	Steve Byrnes	↑	DONE	Done	10/Jun/18	14/Jun/18	



The simplest JQL query is an empty string, as shown here. This means that we are not being selective about which issues that we see, so we see them all. You can see that there are 29 issues in this Jira account, assuming that the current user has permission to see all of the issues. We don't specify how we would like the results to be ordered, so Jira picks the default ordering. It looks like the default order is by the issue key in a descending order.

Topics

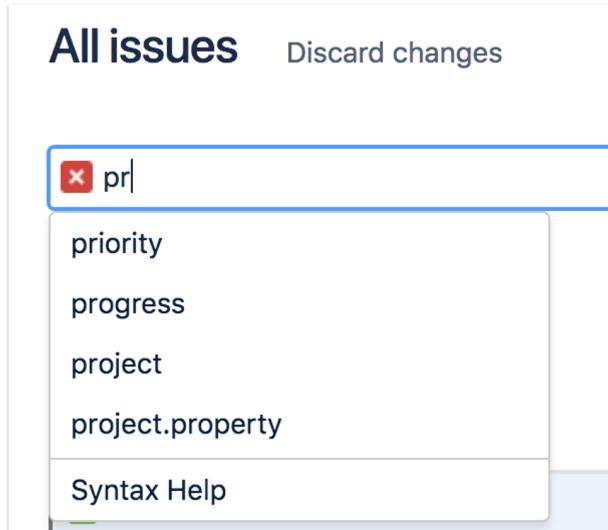
JQL overview

Autocomplete

Functions



JQL with autocomplete-fields



We have the option of creating a JQL query from scratch. Let's create a query that searches for the issues in a project. Project is a field name in a Jira issue and each issue is assigned to a single project. In the advanced search text box, we can begin typing "pr" and Jira will help us with autocomplete suggestions. Autocomplete will show up to 15 matches. This makes writing JQL much easier and helps avoid mistakes. In this case we select the "project" field as the start to the query. Notice that there is a Syntax Help option at the bottom of the list. This will point you to the documentation for advanced search.

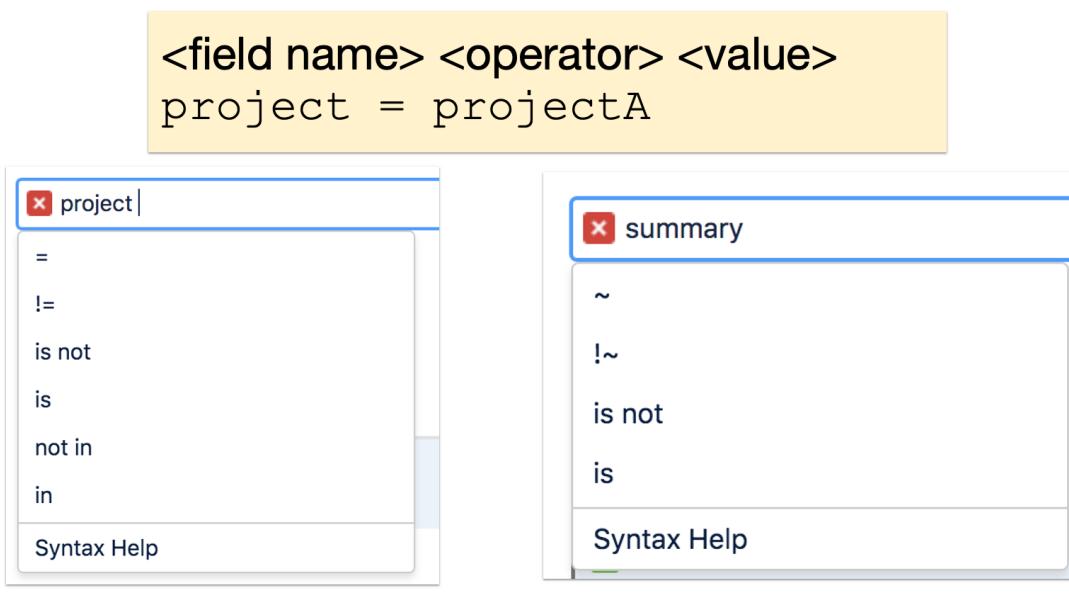
Advanced searching fields reference

The screenshot shows a web browser displaying the Atlassian Jira Software Support documentation. The page title is "Advanced searching - fields reference". The left sidebar contains navigation links such as "Affected version", "Approvals", "Assignee", "Attachments", "Category", "Comment", "Component", "Created", "Creator", "Custom field", "Customer Request Type", "Description", "Due", and "Environment". The main content area includes a table for the "affectedVersion" field and a note about supported operators.

Syntax	affectedVersion
Field Type	VERSION
Auto-complete	Yes
Supported operators	=, !=, >, >=, <, <= IS, IS NOT, IN, NOT IN <small>Note that the comparison operators (e.g. >>) use the version order that has been set up by your project</small>

In the previous example, we knew that we were searching for the project field. If you are unsure of which field to search for, or to get more information related to the field, such as supported operators and functions, search the web for the advanced searching fields reference documentation from Atlassian.

Operator autocomplete



13

An operator is placed between the field name and the value. In this example, the operator is the equals sign.

You can use autocomplete to see which operators that you can use for a specific field name. In this example, we have entered the project field name and then pressed the space bar. Autocomplete provides a list of choices for the operators.

In this example, we have entered the summary field name and then pressed the space bar. Notice that Jira provides a different list of operators for the summary field.

For example, the project field accepts the equals sign as an operator and the summary field accepts the contains operator, which is represented by a tilde (dil-duh). Jira's autocomplete provides the acceptable operators based on the expected type of field value. The project field expects a project name and the summary field expects a free form of text.

Boolean operators

- **AND**
- **OR**
- **NOT**

```
assignee = currentUser() AND status = "In Progress"

status = "Selected for Development" OR status = "In Progress"
status in ("Selected for Development", "In Progress")

NOT status = Backlog
status != Backlog

find unresolved issues in all projects except SampleA
resolution = Unresolved AND NOT project = SampleA
```



Boolean operators are used to either combine or negate clauses. The AND and OR boolean operators are used to combine multiple clauses, allowing you to refine your search.

For example, the first query finds all issues assigned to the current user with a status of "In Progress".

The second query finds all issues with a status of "Selected for Development" OR "In Progress". The third query is equivalent to using the in operator.

The NOT operator is used to negate one or more clauses.

For example, the fourth query finds issues that do not have a status of "Backlog". This is equivalent to using the not equals operator.

In the final example, we find unresolved issues in all projects except for the SampleA project. Notice that we are combining AND and NOT boolean operators. You could do the same this with OR and NOT boolean operators.

Advanced searching- operators reference

The screenshot shows a web browser displaying the Jira Core Support Documentation. The page title is "Advanced searching - operators reference". The left sidebar contains navigation links like "Atlassian Support", "Jira Core", "Documentation", and "Advanced searching". The main content area starts with a section titled "EQUALS: =". It explains that the "=" operator compares the value of a field on its left with one or more values or functions on its right. Below this, there's a note about using multiple "=" statements with the AND operator. A "Examples" section lists two bullet points: "Find all issues that were created by jsmith" with the query "reporter = jsmith" and "Find all issues that were created by John Smith" with the query "reporter = "John Smith"". To the right of the content area, a sidebar titled "On this page" lists various search operators: EQUALS: =, NOT EQUALS: !=, GREATER THAN: >, GREATER THAN EQUALS: >=, LESS THAN: <, LESS THAN EQUALS: <=, IN, NOT IN, CONTAINS: ~, DOES NOT CONTAIN: !~, IS, IS NOT, and WAS.

15

The advanced searching operators reference documents the details related to operators. We will briefly go over operators in this video, but this is mostly to provide you awareness of what is possible with operators in queries. The operators reference is very helpful as you are writing queries.

Executing the query

The screenshot shows the Jira search interface with the query `project = projectA`. The results table displays three issues:

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
	PROJ-3	add feature 3	Steve Byrnes	Steve Byrnes	↑	DONE	Done	18/Jun/18	26/Jun/18	...
	PROJ-2	add feature 2	Steve Byrnes	Steve Byrnes	↑	BACKLOG	Unresolved	18/Jun/18	26/Jun/18	
	PROJ-1	add feature 1	Steve Byrnes	Steve Byrnes	↑	DONE	Done	18/Jun/18	24/Jun/18	

The search bar at the bottom is labeled "the associated basic search" and contains the query `projectA`.

When we complete the query and press Enter, the search is performed. Here we can see that there are three issues in project A. At this point, if we click on the Basic link, we can see that project A is selected in the dropdown list. This is equivalent to the JQL that we just created.

Topics

JQL overview

Autocomplete

Functions



Search clauses

```
<field name> <operator> <field value>
project = projectA
```

```
<field name> <operator> <function>
assignee = currentUser()
```



You usually only want to return a subset of issues from a search. A search clause is used to limit or filter the issues that are returned in the results. The basic structure of the clause is a field name followed by an operator followed by a field value. We have seen an example where we are selecting all issues that have a project field value of "projectA".

Instead of directly providing a field value in the search clause, you can provide a function. A function is a small program that Jira calls before the query is executed. The result of calling the function is then substituted in the query. As an example, this clause will search for issues that have been assigned to the currently logged in Jira user. A big advantage of using functions is that you are not hard-coding information. This same query can be used by all users to view their issues.

Advanced searching- functions reference

The screenshot shows a web browser displaying the Atlassian Confluence page for "Advanced searching - functions reference". The page is part of the "Jira Core Support" documentation. The URL is https://confluence.atlassian.com/jiracorecloud/advanced-searching-functions-reference-765593719.html. The page content includes a brief introduction to JQL functions, a note about unassigned issues, and a list of available functions. The "On this page" sidebar lists functions like approved(), approver(), breached(), cascadeOption(), closedSprints(), completed(), componentsLeadByUser(), currentLogin(), currentUser(), earliestUnreleasedVersion(), elapsed(), endOfDay(), and endOfMonth().

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 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.

approved()

Only applicable if Jira Service Desk is installed and licensed.

Search for requests that required approval and have a final decision of approved.

On this page

- approved()
- approver()
- breached()
- cascadeOption()
- closedSprints()
- completed()
- componentsLeadByUser()
- currentLogin()
- currentUser()
- earliestUnreleasedVersion()
- elapsed()
- endOfDay()
- endOfMonth()

The Jira advanced searching functions are documented so that you can use them if and when you need them. Do a search for advanced searching- functions reference to find it on Atlassian's web site. You can then find the details related to any functions that interest you.

Time-based functions

- `startOfDay()`
- `startOfWeek()`
- `startOfMonth()`
- `startOfYear()`
- `endOfDay()`
- `endOfWeek()`
- `endOfMonth()`
- `endOfYear()`
- `now()`
- `currentLogin()`
- `lastLogin()`

Issues created since the start of today

`created > startOfDay()`



Here is a list of the advanced search functions that are related to time and date. There are functions that specify the start of a time period and end of a time period. The now function represents the current time. The currentLogin function results in the time that the current user logged into this session and the lastLogin function results in the time that the current user logged into the previous session. As an example, this query will find issues created since the start of today.

Time unit qualifier

(+|-)nn(y|M|w|d|h|m)

Tip: Use basic search to
create the query

created in the last 2 days (48 hours):

created > -2d

created since the start of day 2 days ago:

created > startOfDay (-2d)

created since the 15th of this month

created > startOfMonth (+14d)



If the field is related to dates, you can add what's called a time unit qualifier to specify relative dates as field values. This is a specially formatted string that Jira will replace with an actual date value before running the query. In this example, we are searching for issues created in the last two days.

The time unit qualifier is handy because it is simple, and the query doesn't have a hard-coded date.

At the top, you see the syntax that can be used to specify time units. It starts with a plus sign, a vertical bar and a minus sign in parentheses. The parentheses means that what is inside is optional. The vertical bar means or. If you add a minus sign to the string, you are searching back in time. Using the plus sign, or leaving off the sign altogether, means that you are searching forward in time. You can see that this query uses the minus sign, so this query is searching for issues that were created 2 days ago or later.

The nn in the string represents numeric digits. In this string, the numeric digit is 2.

Following the numeric digits is the optional time period unit. You can specify y for a year, capital M for a month, w for a week, d for a day, h for an hour or lowercase m for a minute. In this example, we specified d, so this query is searching for issues created within the last 2 days. We could change that d to a w, for example, to search for issues that were created in the past 2 weeks. If you leave off the units, Jira will assume a logical default unit, which depends on the circumstances.

You can also use time unit qualifiers as arguments to a function. An argument is a value that is passed to the function to change its behavior. If the trailing parentheses of a function call are empty, you are not passing any arguments and the function is

called with its default behavior. In this example, we add an argument to change the function so that we are searching for issues created since the start of day two days ago. If today is Wednesday, this would be searching for issues created Monday or later.

At the bottom is an example of finding issues created the since the 15th of this month, assuming we are currently in the second half of the month. Notice that the function is related to months, but we are passing an argument related to days. The time periods do not need to match.



Takeaways

- A JQL query is behind all basic and advanced searches
- Leverage basic queries and autocomplete to simplify creating JQL queries
- JQL queries may select subsets of issues and/or order query results
- Functions can be used to avoid hard-coding values in a search clause
- Time unit qualifiers (yMwdhm) can be used with date-related values



Tasks

JQL

- Create a basic search and view the JQL query
- Create JQL queries using autocomplete
- Use functions as values



9

Filters



What will you learn?



- Create filters
- Describe board filters
- Use quick filters



Topics

Filters

Board filters

Quick filters



Filters

The screenshot shows the Jira Software interface. On the left, a sidebar titled "Jira Software" lists various filters: Issues and filters, Search issues, STARRED, My custom filter (which is highlighted), OTHER, My open issues (which is also highlighted), Reported by me, All issues, Open issues, Done issues, Viewed recently, Created recently, Resolved recently, and Updated recently. To the right, a main panel titled "My open issues" displays a table of issues. The table has columns for T (Ticket), Key, Summary, Assignee, and Reporter. The issues listed are:

T	Key	Summary	Assignee	Reporter
	PROJ-3	add feature 3	Steve Byrnes	Steve Byrnes
	PROJ-2	add feature 2	Steve Byrnes	Steve Byrnes
	SAM-11	SAM-10 / Update task status by dragging and dropping from column to column >> Try dragging this task to "Done"	Steve Byrnes	Steve Byrnes
	SAM-7	SAM-6 / This is a sample task. Tasks are used to break down the steps to implement a user story	Steve Byrnes	Steve Byrnes
	SAM-14	As a user, I can find important items on the board by using the customisable "Quick Filters" above >> Try clicking the "Only My Issues" Quick Filter above	Steve Byrnes	Steve Byrnes
	SAM-13	As a developer, I can update details on an item using the Detail View >> Click the "SAM-13" link at the top of this card to open the detail view	Steve Byrnes	Steve Byrnes

If you open the global issue navigator, you are presented with search-related tabs, as shown here. Each of these tabs is a filter. Filters are saved searches that can be exposed through user interface elements. Filters are a handy way to execute common searches. If you click on the tab, the search is executed, and you can see the query that was used for the search.

In addition to the default filters that Jira provides, you can create custom filters, such as the "My custom filter" that was created here.

Save a search

The screenshot shows the Jira Software interface. On the left, there's a sidebar with a dark blue background containing navigation links like 'Issues and filters', 'Search Issues', and various 'OTHER' filters such as 'My open issues', 'Reported by me', etc. The main area is titled 'Search' and has a 'Save as' button. It shows a search query with filters: 'Project: All', 'Type: All', 'In Progress', 'Current User', 'Contains text', 'More', and 'Advanced'. Below this, it says '1-2 of 2' and displays two issues in a table:

T	Key	Summary	Assignee	Reporter	P	Status
BUG	SAM-11	SAM-10 / Update task status by dragging and dropping from column to column >> Try dragging this task to "Done"	Steve Byrnes	Steve Byrnes	↑	IN PROGRESS
BUG	SAM-10	As a developer, I can update story and task status with drag and drop (click the triangle at far left of this story to show sub-tasks)	Steve Byrnes	Steve Byrnes	↑	IN PROGRESS



Saving a search creates a filter. Start by creating and executing a query in either basic or advanced search, then click on the Save as link to begin the process of saving the filter. In this example, we have created a search that selects issues that have a status of In Progress and that the current user owns. We then click on the "Save as" link to create the filter.

Save the filter

The screenshot shows the Jira Software interface. On the left, there's a sidebar with navigation links like 'Issues and filters', 'Search issues', and various 'OTHER' categories such as 'My open issues', 'Reported by me', etc. The main area is titled 'Search' with a 'Save as' button. A modal window titled 'Save Filter' is open, prompting the user to enter a 'Filter Name*' (marked with a red asterisk). The input field contains the text 'My in progress'. At the bottom of the modal are 'Submit' and 'Cancel' buttons.



We will name this filter "My in progress".

View all filters

The screenshot shows the Jira Software interface with a sidebar on the left and a main content area on the right.

Left Sidebar:

- Jira Software logo
- Issues and filters
- STARRED**
 - My in progress
- OTHER**
 - My open issues
 - Reported by me
 - All issues
 - Open issues
 - Done issues
 - Viewed recently
 - Created recently
 - Resolved recently
 - Updated recently
- View all filters** (button)

Main Content Area:

Filters

Owner: Shared with: Popularity

Name	Owner	Shared with	Popularity
Filter for PRJ board	Steve Byrnes	project: projectB	0
Filter for PROJ board	Steve Byrnes	project: projectA	0
Filter for SAM board	Steve Byrnes	project: SampleA	0
My in progress	Steve Byrnes	Private filter	1

Prev 1 Next

A context menu is open over the last row ('My in progress') with the following options:

- Manage subscriptions
- Copy filter
- Edit filter details
- Delete filter

Once you have created the filter, it will show in global issue navigator under the "starred" category. Also, if you click on View all filters, you can see your newly created filter in the list. By default, the filter is a private filter, meaning that it is only accessible to you. You can click on the more icon to the right of the filter to change metadata related to the filter. Select Edit filter details to edit it.

Edit filter metadata

Edit Current Filter ⓘ

Name*
My in progress

Description

Favorite
★

Shares
Not shared

Project: projectA

All

Shared with everyone with permission to browse the 'projectA' project

Save Cancel



On the Edit Current Filter screen, you can change the name of the filter, add a description, add or remove it from your favorite filters, and share the filter with others. Here we are sharing the filter with the members of the project A team. They will see the filter in their sidebar. Filters are application-wide entities in Jira. That is why they are configured in the application-level sidebar. The extent to which a filter is visible to users depends on the extent that they are shared using this screen. Depending on the permissions of the user creating the filter, they can choose to keep the filter private, share it with only certain projects or groups of people, or expose the filter to all users of the account.

Edit filter query

The screenshot shows the Jira Software interface for editing a filter query. On the left, a sidebar menu lists various issue categories: STARRED, OTHER, My open issues, Reported by me, All issues, Open issues, Done issues, Viewed recently, Created recently, Resolved recently, and Updated recently. The 'My in progress' option is selected and highlighted in blue. The main area displays a search results page titled 'My in progress' with the URL `status = "In Progress" AND assignee in (currentUser()) ORDER BY updated DESC`. The results table shows three items:

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
	PROJ-1	add feature 1	Steve Byrnes	Steve Byrnes		IN PROGRESS	Unresolved	18/Jun/18	30/Jun/18	...
	SAM-11	SAM-10 / Update task status by dragging and dropping from column to column >> Try dragging this task to "Done"	Steve Byrnes	Steve Byrnes		IN PROGRESS	Unresolved	14/Jun/18	18/Jun/18	
	SAM-10	As a developer, I can update story and task status with drag and drop (click the triangle at far left of this story to show sub-tasks)	Steve Byrnes	Steve Byrnes		IN PROGRESS	Unresolved	10/Jun/18	10/Jun/18	



To edit the query for a filter, execute the original filter so that you can view the query. Modify and execute the query. When the query is modified, you will see the Edited indication next to the filter name and can click the Save link to overwrite the existing query.

Topics

Filters

Board filters

Quick filters



Board filters

- Every board has a filter that defines the issues shown on the board
- You can edit the board's filter
- If you create a board, you must assign it a filter

Filter

[Saved Filter](#)

[Filter for PRJCT board](#)

[Edit Filter Query](#)

[Filter Query](#)

`project = PRJCT ORDER BY Rank ASC`



A typical board for a project contains a filter that only shows issues for the project using a "project = xxx" clause. There may be other filters (such as sub-filters) associated with the board, further limiting the issues shown on the board.

Example board filter

A board filter can be used to show issues from multiple projects on a single board

Filter

Saved Filter

Two Projects

[Edit Filter Query](#)

Shares

No shares

[Edit Filter Shares](#)

Filter Query

project in (PRJCT, PRJ) ORDER BY Rank ASC

Kanban board



Topics

Filters

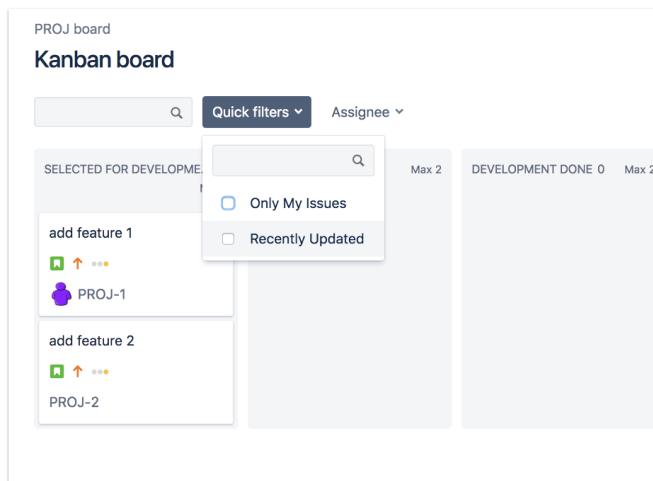
Board filters

Quick filters



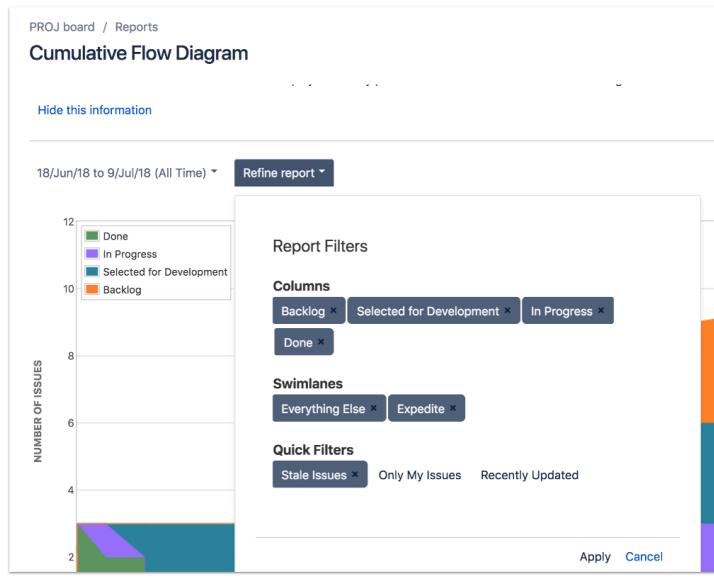
Filtering a board: quick filters

Further filter issues displayed on a board



Quick filters are a way to further filter issues displayed on a board. By default, all the issues are displayed. You can see that there are two default quick filters. Check the Only My Issues box to limit the board to displaying issues in which you are the assignee. Check the recently updated checkbox to show only issues that have been updated in the last day. If you have a lot of quick filters, you can use the search box to find your filter. Here we only have the default quick filters, so the search box is not too helpful.

Quick filters and reports



A board's quick filters can also be used to refine reports. Here we are viewing the cumulative flow diagram report. If we click on the "refine report" dropdown, we see that we can deselect columns or swimlanes to hide them from the report. We will discuss swimlanes a little later. We could further filter the issues that are being reported on by clicking on one or more quick filters to enable it. They are all not enabled by default. Be sure to click "Apply" to see the changes to the report.



Takeaways

- Filters are saved searches that can be exposed through user interface elements
- Every board has a filter that defines the issues shown on the board
- Quick filters are saved searches that are used to further limit the issues displayed on a board or in reports



Tasks

Filters

- Explore default filter queries
- Create a starred filter
- Explore and create quick filters



10

Epics



What will you learn?



- Describe epics
- Work with epics
- Manage epics in the backlog



Topics

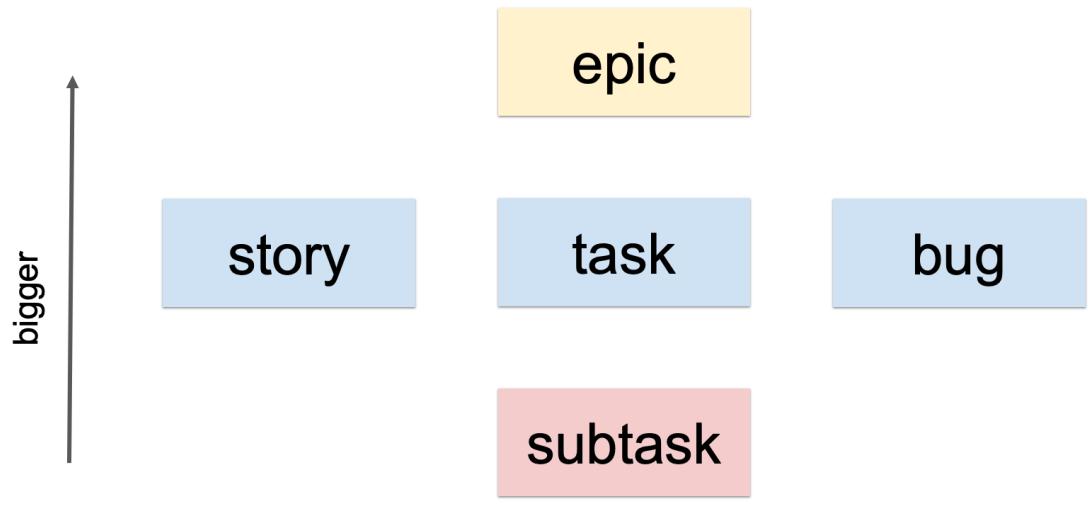
[Epics overview](#)

[Working with epics](#)

[Epics in the backlog](#)



Jira's issue type hierarchy



We have discussed stories, tasks, bugs, and subtasks. An epic is an issue type that has a bigger scope than all of these. How they are actually used is up to the team.

Epic

- A large issue
- Can contain other issues
- Child issues can span multiple iterations, projects, teams and boards
- Can be a placeholder for many stories

Create issue

Project* projectA (PROJ)

Issue Type* Epic

Epic Name* Big Feature A

Summary* add big feature A

Reporter* Steve Byrnes

Component/s None

Create another Create Cancel



An epic is a large issue.

An epic can contain issues. They can contain stories, tasks, bugs and custom issue types.

Child issues of an epic can span multiples iterations, projects, teams and boards.

It can serve as a placeholder for many stories. For example, an epic might be to create an iPhone version of an app. You then would break this into stories when the team is available to actually work on it.

Why epics?

- Organization of work
- Span multiple iterations and projects
- Simplifies backlog (one issue)



Epics are useful because they provide an organization of the team's work, rather than grouping unrelated stories.

They are also useful if the work spans multiple iterations.

Also, have one placeholder story to represent a lot of work can simplify the backlog.

You don't want to create detailed plans for an epic too early.

Topics

Epics overview

Working with epics

Epics in the backlog



Creating an epic

The screenshot shows the 'Create issue' dialog in Jira. At the top, there are buttons for 'Import issues' and 'Configure fields'. Below that, a 'Project' dropdown is set to 'projectA (PROJ)'. The main form area starts with an 'Issue Type' field, which is currently set to 'Epic' (indicated by a lightning bolt icon). A tooltip message says, 'Some issue types are unavailable due to incompatible field configurations.' Below this is an 'Epic Name' field containing the text 'Big Feature A'. A placeholder text 'Provide a short name to identify this epic.' is visible below the input field. At the bottom of the form, there is a 'Component/s' section with 'None' selected. On the right side of the form, there are buttons for 'Create another', 'Create' (in blue), and 'Cancel'.



You can create an epic by selecting the Epic issue type when creating an issue. Here we have created an epic named Big Feature A.

Viewing the issues of an epic

The screenshot shows a Jira epic issue titled "add big feature A" (PROJ-12). The epic has a status of "Backlog" and is unassigned. It contains one child story, "PROJ-13 epic story A", which is also in the "Backlog" status. The story was created 7 minutes ago and updated 16 seconds ago. The reporter is Steve Byrnes.

Issue	Status	Reporter
PROJ-12 add big feature A	Backlog	Steve Byrnes
PROJ-13 epic story A	Backlog	Steve Byrnes

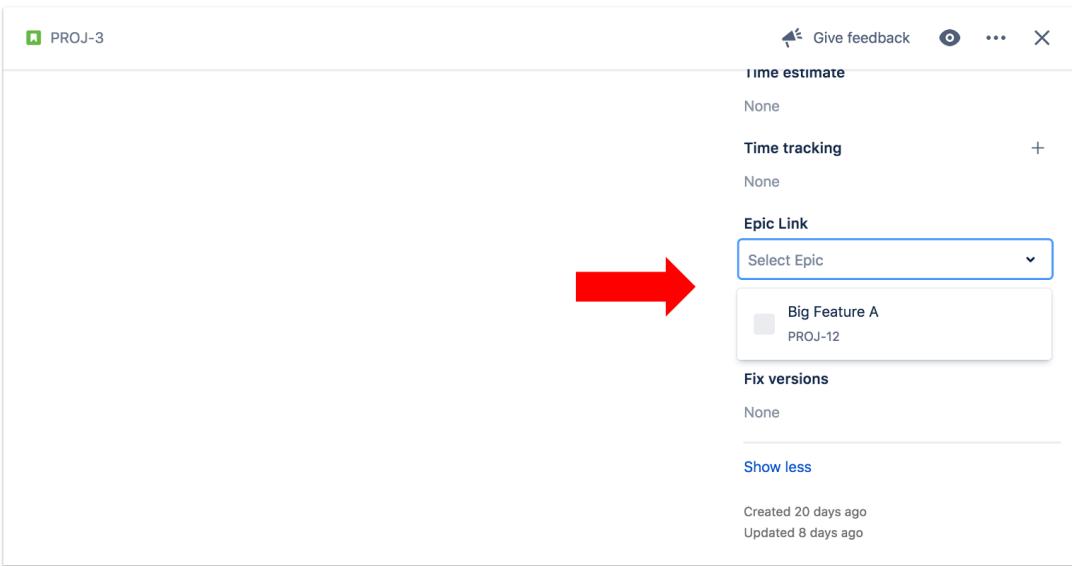
To view the issues of an epic, you can open the epic issue.
Issues that belong to this epic are shown.

Epic link

The screenshot shows a Jira issue page for 'epic story A'. At the top left is a green square icon labeled 'PROJ-13'. On the right are buttons for 'Give feedback', three dots, and a close button. Below the icon is the title 'epic story A' and a row of edit, copy, and other icons. A text input field says 'Add a description...'. Underneath is a 'Comments' section with a placeholder 'Add a comment...' and a user icon. To the right are fields for 'Status' (set to 'Backlog'), 'Assignee' (set to 'Unassigned'), and 'Labels' (containing 'Epic Link' and 'Big Feature A' in a box). Below these is a 'Reporter' field with a user icon and 'Steve Byrnes'. A 'Show more' link is at the bottom. A blue triangle icon is in the bottom right corner of the main panel.

If you view an issue belonging to an epic, you will see the epic in the Epic Link field. You can click on the link to view the epic.

Adding an existing issue to an epic



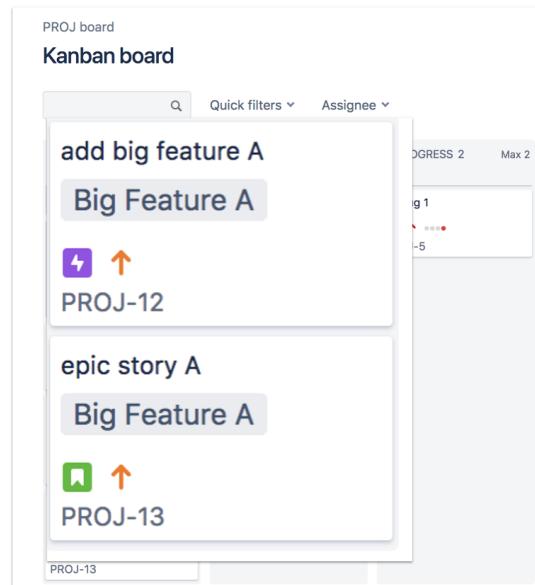
You also can add an existing issue to an epic at any time using the Epic Link dropdown.

Searching for issues of an epic

The screenshot shows a Jira search interface. At the top, there is a search bar with the query: "Epic Link" = PROJ-12 ORDER BY lastViewed DESC. Below the search bar, there is a dropdown menu with options: "Big Feature A - (PROJ-12)" and "Syntax Help". The main search results area has columns: T, Key, and Summary. One result is listed: PROJ-13 epic story A. At the bottom left, it says 1-1 of 1. On the right side of the results area, there is a blue triangle icon.

An epic is an issue type in Jira and, like all issue types, has an issue key. If an issue is part of an epic, its epic link field contains the issue key of the epic issue, which can be thought of as its parent. Here we are searching for all issues that are part of this epic. We can use Jira's autocomplete to help create this query.

Epic labels on a board



The kanban board can show the epic issue itself and any issues that belong to it are labeled.

Here you can see the epic issue, and a story that belongs to the epic.

Epic-based swimlanes

The screenshot shows a Jira Kanban board titled "PROJ board" with a section labeled "Kanban board". At the top, there are search and filter options: "Quick filters" and "Assignee". Below these are two columns: "SELECTED FOR DEVELOPMENT 5" and "IN PROGRESS 1".

The board displays several issues:

- A collapsed section titled "Big Feature A 1 issue" contains one card: "epic story A" under "Big Feature A". This card has a green upvote icon, a purple "PROJ-13" label, and a purple user icon.
- A collapsed section titled "Issues without epics 7 issues" contains two cards: "add feature 2" and "add feature 1". Both cards have green upvote icons, purple "PROJ-2" and "PROJ-1" labels, and purple user icons.



Here we see an epic-based swimlane. The issues of the epic are separated from the rest of the issues on the board.

Topics

[Epics overview](#)

[Working with epics](#)

[Epics in the backlog](#)



Creating an epic from a backlog

The screenshot shows the Jira interface for a project named 'projectB'. On the left, there's a sidebar with various project management options like 'Backlog', 'Active sprints', and 'Reports'. The main area is titled 'PRJ board' and 'Backlog'. It features a search bar, 'Quick filters', and 'Assignee' dropdowns. A red arrow points to the 'Create epic' button, which is located under the 'EPICS' tab in the header. Below the header, there's a section for 'All issues' and a summary for 'PRJ Sprint 2' (1 issue). At the bottom right, there's a '+ Create issue' button.

You can click "Create epic" under the Epics tab of the backlog.

Epics panel

The screenshot shows the Jira interface for the 'PRJ board' project. On the left, a sidebar menu includes options like 'Backlog' (which is selected), 'Active sprints', 'Reports', 'Releases', 'Issues and filters', 'Pages' (marked as 'NEW'), 'Components', 'Add item', and 'Settings'. The main area is titled 'Backlog' and shows a summary for 'PRJ Sprint 2' which contains 1 issue. It lists 'Big Feature B' under 'PRJ-6 add feature B' with status: Issues (0), Completed (0), Unestimated (0), and Estimate (0). A link 'Create issue in epic' is visible. Below this, there's a section for 'Backlog' with 0 issues and a '+ Create issue' button.

You can use the epics panel in the backlog to monitor the current status of the epic. Notice the create issue in epic link. You can start the process of creating an issue in the epic here.

Mark an epic as done

The screenshot shows the Jira Backlog board for the projectB Software project. The left sidebar shows navigation options like PRJ board, Backlog (which is selected), Active sprints, Reports, Releases, Issues and filters, Pages (NEW), Components, Add item, and Settings. The main area displays the backlog with a summary for PRJ Sprint 2 (1 issue) from 30/Jun/18 to 07/Jul. A specific epic, "Big Feature B" (PRJ-6 add feature B), is listed with 1 issue, 0 completed, 1 unestimated, and 0 estimate. A context menu is open next to the epic name, showing options: add feature 3, Color (with a palette of colors), Edit name, View epic details, and Mark as Done (which is highlighted with a red box).

You can mark an epic as done in the backlog. Select mark as done from the dropdown next to the epic name.

An epic marked as done is removed from the epics panel

The screenshot shows a Jira Backlog board titled "PRJ board" with the section "Backlog". At the top, there is a search bar, "Quick filters", and "Assignee" dropdowns. Below the header, there are two tabs: "EPICS" (selected) and "Create epic". A "VERSIONS" sidebar lists "All issues". The main area displays two sections: "PRJ Sprint 2" (1 issue, 30/Jun/18 3:54 PM - 07/Jul/18 3:54 PM) and "Backlog" (1 issue). Each section has a "add feature" button. A "Create issue" button is also present at the bottom.



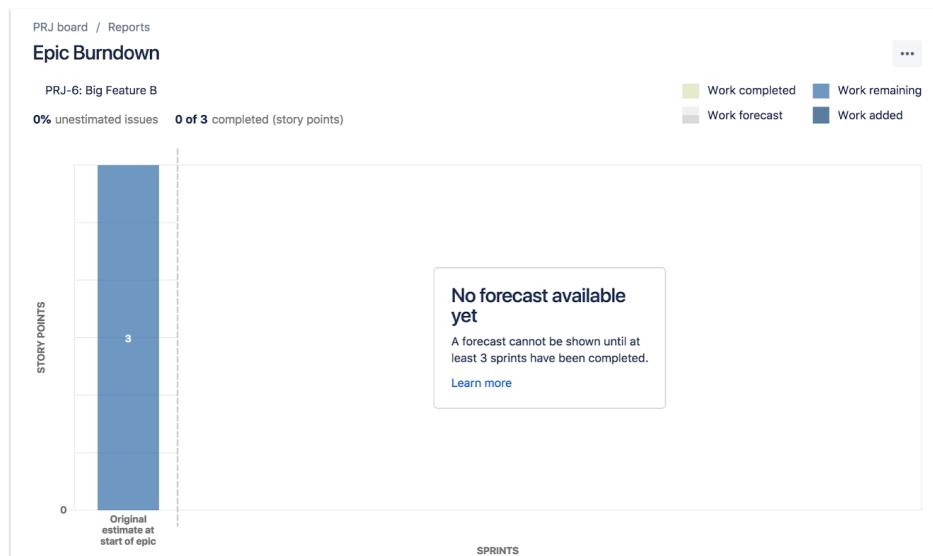
When you mark an epic as done, it no longer shows up in the epics panel of the backlog. The epic status field for the epic issue is set to a value of done.

Epic report



Under the reports tab, you can view the Epic Report. This shows the current status of the epic, as well as a list of the issues in the epic.

Scrum- epic burndown chart



You can view the epic burndown chart to see your progress on the epic as you complete sprints.



Takeaways

- An epic is a large issue of issue type “epic” that may contain other issues
- The “epic link” field is used to associate issues with an epic
- Epics can be shown on boards or in backlogs



Tasks

Epics

- Create an issue of type “epic”
- Add issues to the epic
- View swimlanes by epic
- View the epic in the kanban backlog
- Complete an epic



11

Dashboards



What will you learn?



- Describe dashboards
- Configure a dashboard
- Display a dashboard as a wallboard



Visualizing work



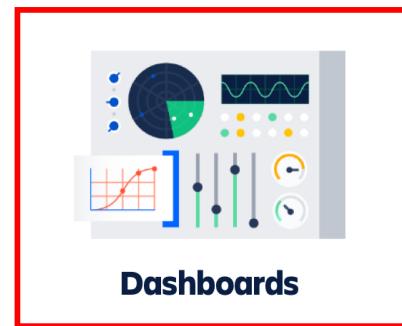
Boards



Search



Reports



There are the main ways to visualize work using Jira. Here we will discuss dashboards.

Dashboards

The dashboard is titled "Alana's dashboard". It contains four gadgets:

- Sprint Health Gadget:** Shows "PROJB Sprint 3 - PROJB board". Overall sprint progress is at 0% (Story Points). 5 days left. Status: 0% Time elapsed, 0% Work complete, 0% Scope change, 0 Blocker, 0 Flagged.
- Activity Stream:** Shows activity from "Your Company JIRA". Recent updates include: "Alana Grant updated the Sprint of PROJB-4 - add item 8" (1 minute ago), "Alana Grant updated the Sprint of PROJB-5 - add item 7" (1 minute ago), "Alana Grant updated the Story Points of PROJB-7 - add item 9" (1 minute ago), "Alana Grant created PROJB-7 - add item 9" (2 minutes ago), and "Alana Grant updated the Story Points of PROJB-6 - add item 8" (2 minutes ago).
- Sprint Burndown Gadget:** Shows "Sprint Burndown Gadget" for Nov 3 to Nov 9. Story Points remaining: 6 (Nov 3), 5 (Nov 4), 4 (Nov 5), 3 (Nov 6), 2 (Nov 7), 1 (Nov 8), 0 (Nov 9). Legend: Guideline (blue bar) and Non-Working Days (red bar).
- Assigned to Me:** Shows issues assigned to Alana. Key, Summary, and Priority (P) are listed. Issues include: TIS-57 (Buttons need to be red), TIS-58 (Add Features for Flight School Teachers), TIS-59 (Enhancements for Flight School), PROJ-8 (fix bug 1), PROJ-11 (add big feature A), PROJL-1 (add item 1), PROJL-2 (Create initial release), PROJL-3 (add item 6), and PROJB-5 (add item 7).

- Configurable view of the work of one or more projects
- Can be personal or shared
- Contains gadgets



Dashboards are a way to present a customized view of projects. They can be created for personal use or to share with others on the team.

Dashboards are made up of one or more gadgets. A gadget displays some aspect of the work of the projects. In this example, we see four gadgets. Two are related to sprints, one for an activity stream and another showing issues assigned to the current user.

Types of gadgets

The screenshot shows a 'Search' input field at the top. Below it is a section titled 'CATEGORIES' with the following items:

CATEGORY	Count
All	31
Charts	8
Jira	27
Other	3
Wallboard	7



Gadgets can be divided into categories:

Charts – Displays information visually using pie charts, bar charts, etc.

Jira – Present information about issues. The issues could be filtered by project, team member, status, etc.

Wallboard – The information in these gadgets can be displayed in dashboards, but is optimized for display as a wallboard.

ATLASSIAN Marketplace

Atlassian Marketplace for JIRA

Discover powerful apps compatible with your JIRA version via the Atlassian Marketplace.

The screenshot shows the Atlassian Marketplace interface for JIRA. At the top, there's a search bar, a staff-picked dropdown, and a dropdown menu set to 'Dashboard gad...'. Below these are two app cards:

- Xray Test Management for Jira** by Xpand IT (Top Vendor)
Category: CUSTOM FIELDS | DASHBOARD GADGETS | JIRA SERVICE DESK | TESTING & QA
Rating: ★★★★ (231)
Installs: 6,562
Price: \$2500
Description: Xray is a Jira App for managing quality assurance tests which integrates seamlessly with Jira. Xray supports manual and automated tests and provides reports to track test coverage of your Jira projects. Xray includes a RESTful API out of the box.
- Arsenale Dataplane - Jira Reports** by Arsenale (Top Vendor)
Category: CHARTS & DIAGRAMMING | DASHBOARD GADGETS | PROJECT MANAGEMENT | REPORTS
Rating: ★★★★ (55)
Installs: 1,142
Price: \$1250
Description: Dataplane delivers powerful, intuitive Jira reports and metrics for managers. Dozens of built-in reports; save, share and subscribe to reports; display on dashboards; export to PDF and Excel.

- Contains more dashboard gadgets
- Select “Dashboard gadgets”

Adding gadgets

Add gadget Edit layout ...

Add a gadget

Search

CATEGORIES

Category	Count
All	31
Charts	8
Jira	27
Other	3
Wallboard	7

Activity Stream
By Atlassian • Local
Lists recent activity in a single project, or in all projects.
[Show XML link](#)

Agile Wallboard Gadget
By Atlassian • Local
Displays a board as a Wallboard gadget
[Show XML link](#)

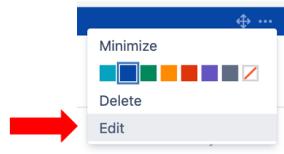
Assigned to Me
By Atlassian • Local
Displays all unresolved issues assigned to me
[Show XML link](#)

Average Age Chart
By Atlassian • Local
Displays the average number of days issues have been unresolved.
[Show XML link](#)

Blue Atlassian logo

Click the Add gadget button to bring up a list of gadgets to add to your dashboard.

Configuring a gadget



The configuration dialog for the 'Sprint Health Gadget' is shown. It includes the following settings:

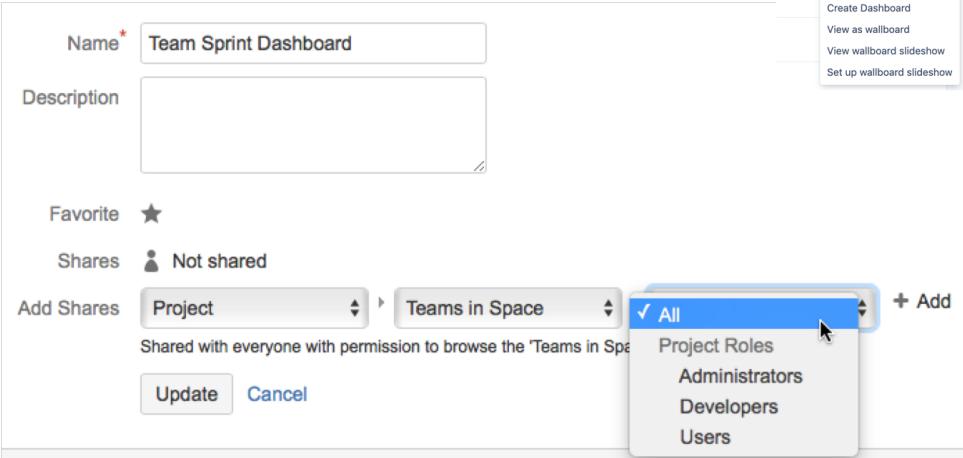
- Board:** PROJB board
- Sprint:** Next Sprint Due (auto)
- Auto refresh:** Update every 15 minutes
- Show options:** Show board name, Show sprint name, Show assignees

At the bottom are 'Save' and 'Cancel' buttons.



Most gadgets can be configured to show specific information. In this example, we are specifying the information related to the Sprint Health Gadget, such as which board to use as the source of data.

Sharing dashboards



The screenshot shows the 'Share Dashboard' dialog in Jira. At the top, there are fields for 'Name' (Team Sprint Dashboard) and 'Description'. Below that is a 'Favorite' button with a star icon. Under 'Shares', it says 'Not shared'. A dropdown menu for 'Add Shares' is open, showing 'Project' and 'Teams in Space' as options. The 'Teams in Space' option is selected, and a sub-menu is displayed with 'All' checked (indicated by a blue highlight and a checkmark). Other options in the sub-menu include 'Project Roles', 'Administrators', 'Developers', and 'Users'. At the bottom of the dialog are 'Update' and 'Cancel' buttons.

Shared with everyone with permission to browse the 'Teams in Space'

Update Cancel

Add gadget Edit layout ...

Copy Dashboard

Edit Dashboard

Share Dashboard

Delete Dashboard

Find Dashboards

Create Dashboard

View as wallboard

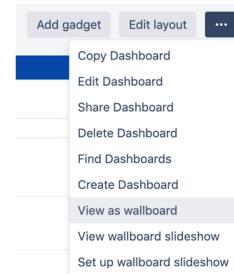
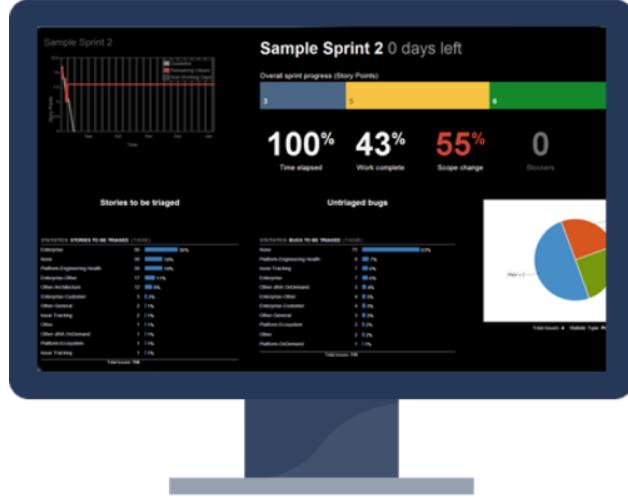
View wallboard slideshow

Set up wallboard slideshow

Dashboards can be configured and used privately. They also can be shared with all or part of the team.

Wallboards

- Turn any dashboard into a wallboard
- Acts as an information radiator



A wallboard is usually a television or monitor that is visible in a room. This radiates project status to the team, increasing the team's shared understanding of the projects.

Takeaways



- Dashboards display the work of projects
- Dashboards can be shared or used personally
- Gadgets display a portion of a dashboard
- Dashboards can be shown as a wallboard to radiate information



Tasks

Dashboards

- Create a dashboard
- Display a dashboard as a wallboard



12

Putting it all Together



Topics

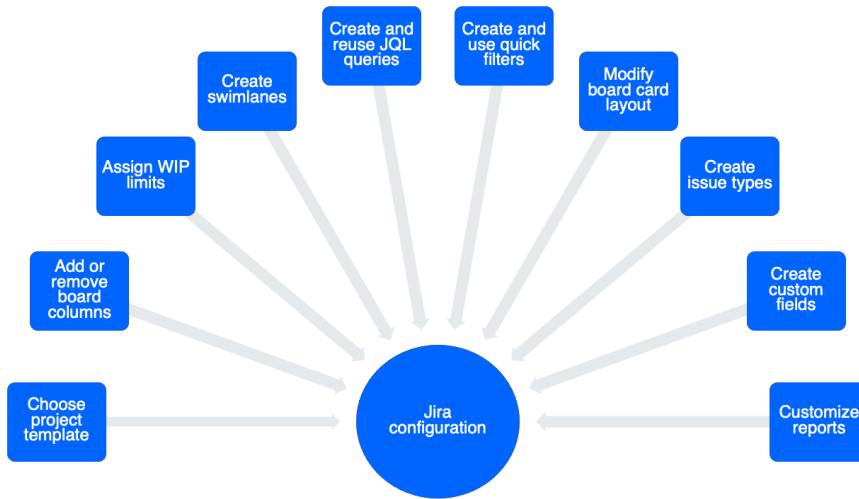
Quick course review

Jira family

Wrap up



Ways to configure Jira to match your team's process



Each agile project has a different setup

- Different members
 - Different processes
 - Different personal preferences
 - In total we have 9 areas that we can configure
-
- This is a continuous process. Agile is very responsive to change. Reconfiguring boards is just one way we can respond to changing needs.
 - We'll be covering all areas in subsequent modules but let's start with the general configuration for now.

Topics

Quick course review

Jira family

Wrap up



Jira Family of Products



Jira Core
Business project management software



Jira Software
Plan, track, and release software



Jira Service Desk
Service desk software for IT teams

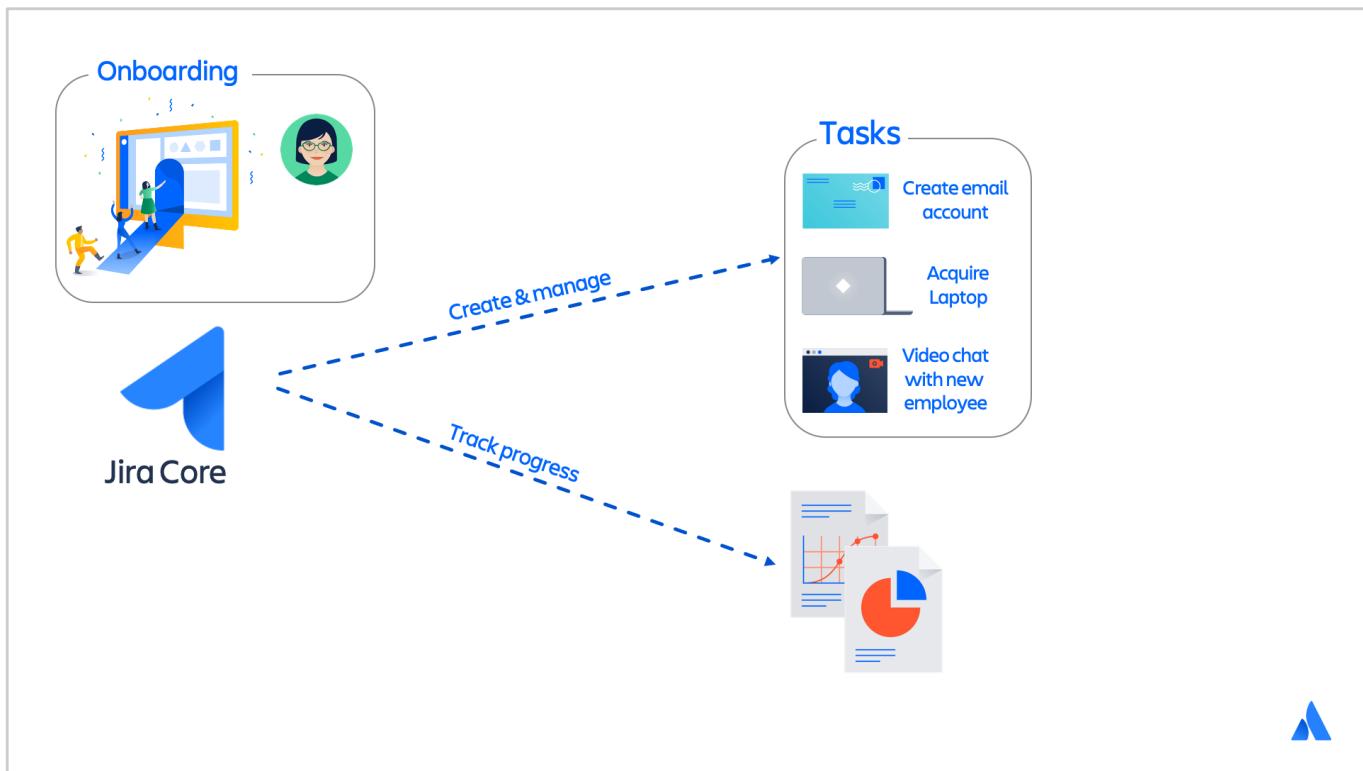


Jira is a family of three products.

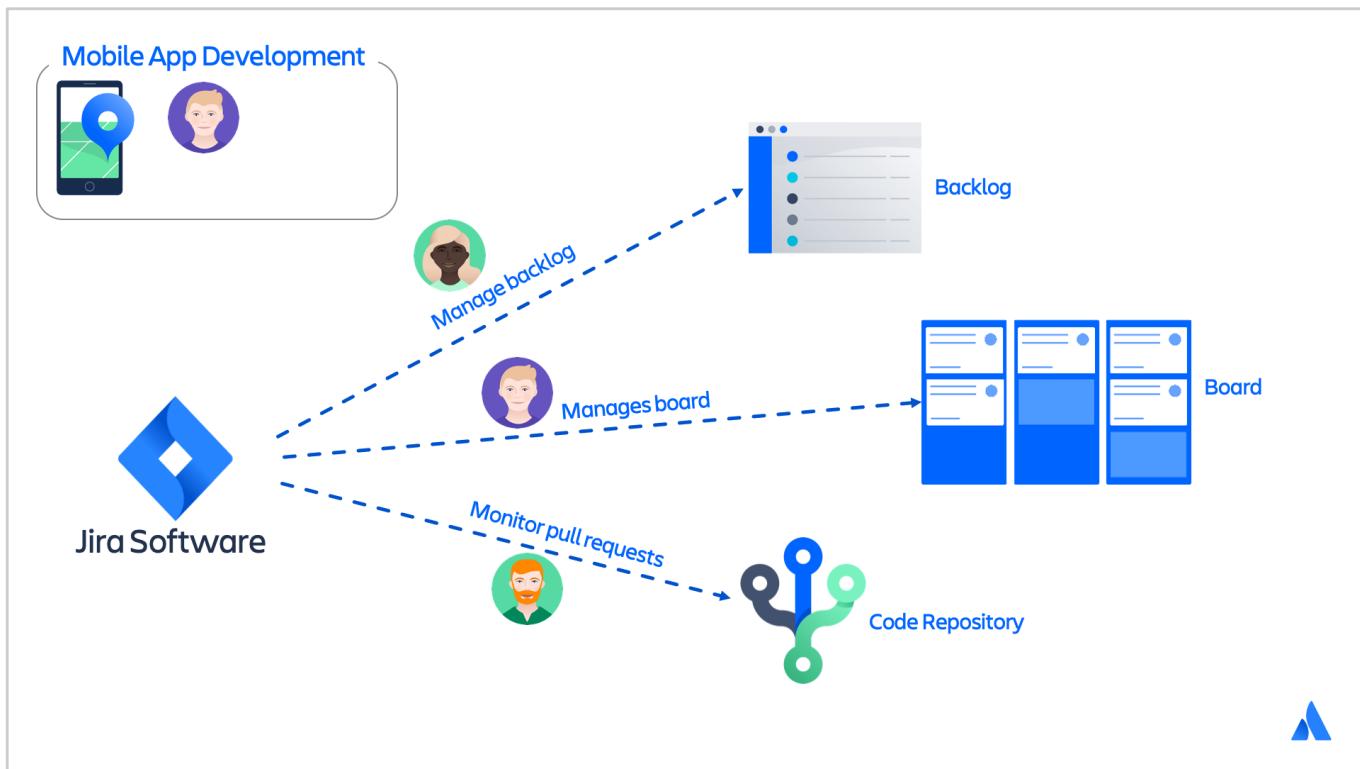
- Jira Core - Used to manage non-software projects and help keep teams organized. Allows you to manage projects, monitor details, and measure performance.
- Jira Service Desk – Extends the core functionality of Jira Core. Used by IT and Customer Service teams to create and run their service desks. Has capabilities, which include self-service, automation, SLAs, and CSAT reporting.
- Jira Software – Extends the core functionality of Jira Core. A software development tool for agile teams that allows them to plan, track, release, and report their progress developing software. Whether your team uses Scrum, Kanban, Scrumban, or something else, Jira Software is designed with Agile teams in mind.

Both Jira Service Desk and Jira Software combine the core functionality provided by Jira Core such as issue tracking, workflows, and reporting, and add their unique functionality that allows you to manage your projects.

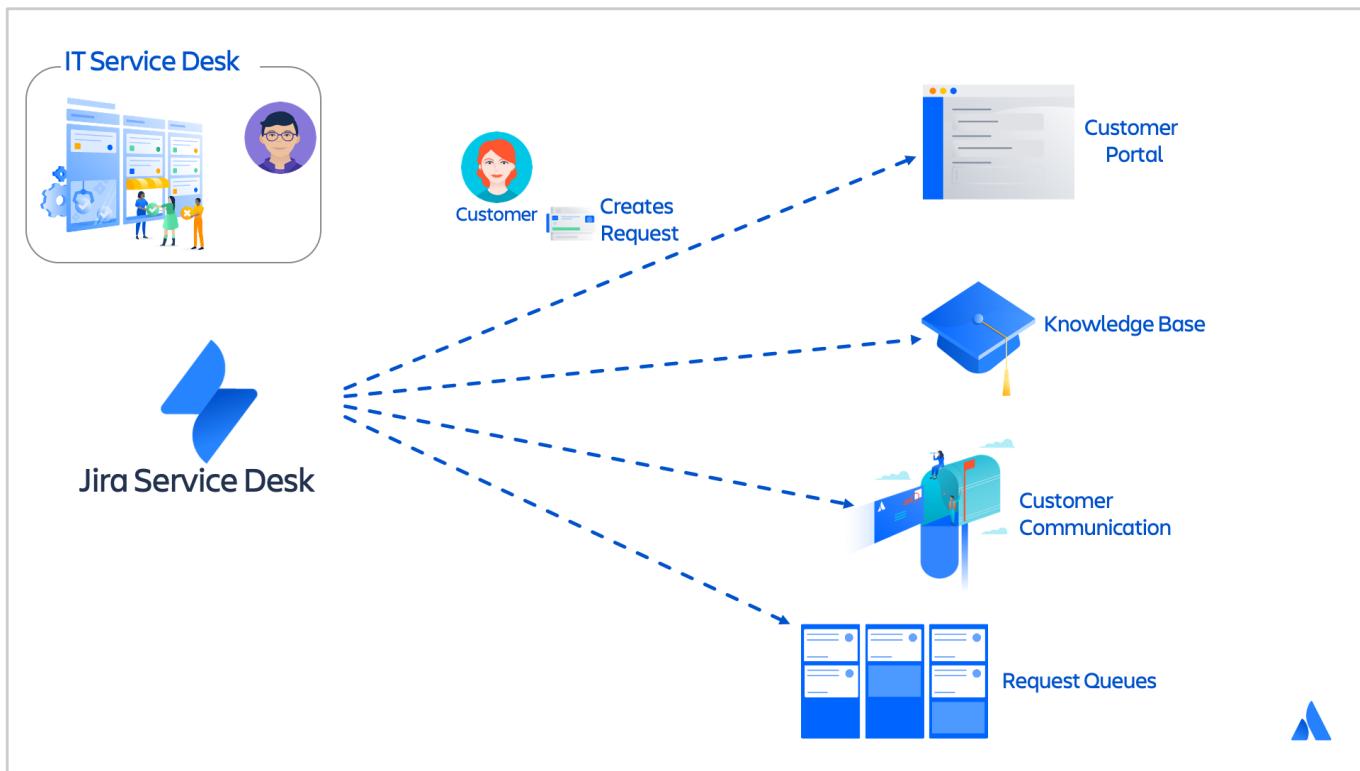
You can use Jira Core alone, a combination of Jira products, or all three Jira products.



Jira Core helps you manage business projects. For example, Cassie manages an HR team, which uses Jira Core to manage their Onboarding project. Within Jira, they can create tasks and manage them. A task can be anything from creating an email address for the new employee, acquiring his/her new laptop, or keeping in touch before their start date. Cassie can get a quick look at the overall statistics to ensure the tasks aren't getting bottlenecked, and that helps keep the project on track

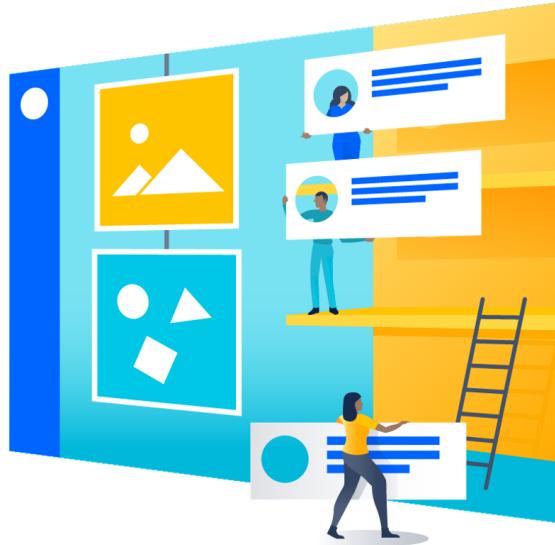


Jira Software is designed to help you manage software projects. For example, Will manages this agile software development team. They are currently adding location functionality to their mobile application. They manage all their work on a board and Will manages the board, which contains everything the team is committed to finishing. Emma, the Product Manager manages the backlog of items that still need to be started. Jira Software allows developers like Kevin to create branches and monitor pull requests and build statuses right from the Jira issues.



Jira Service desk helps you manage your IT service desk. For example, Ryan's team uses Jira Service Desk to manage their internal IT service desk. It allows the Service Desk team members to resolve problems and service requests from other employees throughout the company. Employees create requests using the Customer Portal. The requests immediately show up in the service desk queues. The service desk team members monitor these queues and grab the most urgent requests on which to work. Sometimes, they can resolve a problem immediately from the knowledge base suggestions that automatically appear in the service desk request. They can also communicate with customers and stakeholders.

ATLASSIAN Marketplace



Can't find what you want for a gadget or a report out-of-the-box in Jira? Try the Atlassian Marketplace. There are over 1,000 apps available for just Jira. Someone may have already created what you need. Still can't find what you want? Jira has a rich UI in which a developer can create a custom gadget.

Tasks

(optional) Final project

- Modify a Jira project to match your team's processes



Topics

Quick course review

Jira family

Wrap up



Earn Atlassian Certifications!



Propel your career



Boost your skills



Help your teams do
their best work

atlassian.com/certification



Atlassian skills are in high demand in the job marketplace. The path to Atlassian certification is designed to help you deepen and expand your Atlassian skills, and prepare you to take on your next challenge.

Follow the path to Certification – you'll propel your career, boost your skills and learn new ways to help your teams do their best work.

To find out more, visit atlassian.com/certification.

Training for Jira

Marketplace App



- Interactive tutorials
- Gets teams using Jira quickly
- Covers popular topics and tasks
- Introduces essential concepts
- Showcases in-product demos
- Non-graded quizzes to track learning
- Available for Cloud, Server, Data Center



Training for Jira is a new Marketplace app containing a growing collection of short, interactive tutorials, designed to get teams over the hump and using Jira quickly.

Why Training for Jira?

- Help teams help themselves to training, and reduce the number of Jira questions.
- Training for Jira scales – it's built right into Jira and is available to everyone.
- The tutorials are short and topic-focused; users can choose what to learn and when.
- The app also keeps track of users' progress so they can see where they left off, and so managers can track the progress of their teams.

The Tutorials...

- Cover popular topics and tasks
- Introduce essential concepts
- Showcase in-product demos
- Include non-graded quizzes so you can see if you're getting it
- Have optional voice-over narration and closed captions

Available for Cloud, Server, and Data Center

Learn more

- Search for "Training for Jira" from marketplace.atlassian.com to download a free trial.
- Requires internet connectivity to access content
- Note: If you're using a firewall or have network restrictions where Jira is installed, you may need to whitelist IP addresses to access this app.

Training Credits



A prepaid account for Atlassian product training that gives your business the scale and flexibility to train your employees

Simplified approvals

12-month burn-down balance avoiding repeated purchase approvals

Volume discounts

Up to 20% at volume for teams

Fast access

Reduce purchasing delays and get training quickly



Simplified Approvals

Training credits allow your procurement team to establish a 12-month burn-down balance for Atlassian training. With this simple arrangement, you'll no longer need to process repeated purchase approvals for small transactions.

Volume Discounts

By pre-paying for training as an annual credits program, you can take advantage of volume training discounts to get a more effective return on your training budget.

Fast Access

With training credits, you reduce the purchasing delays for your business users and make it faster to access the training they need.

Want to learn more?

Training & Certification	atlassian.com/university
Training & Certification Online Community	go.atlassian.com/traincertcommunity
General Online Community	community.atlassian.com
Documentation	confluence.atlassian.com
Support	support.atlassian.com
Enterprise Services	atlassian.com/enterprise/services
More Resources	atlassian.com/resources



Join your local Atlassian Community Events (ACE) group

What happens at an ACE?

A diverse group of Atlassian users, both technical and non-technical, come together to share best practices and enjoy food and drink.

How do I join?

ACEs are in over 30 countries around the world. Go to aug.atlassian.com to join a group near you.

No ACE in your city?

You can start one! Go to aug.atlassian.com/leaders to sign up. You can also connect with fellow users on community.atlassian.com



**Please take the survey and
give us your feedback**



Congratulations on completing the
course!

