

Defining a Project

This page tells you how to **add a new project** or **configure an existing project**. You must be a [JIRA administrator](#) to add/configure a project.

A JIRA project is a collection of issues. Your team could use a JIRA project to coordinate the development of a product, track a project, manage a help desk, and more, depending on your requirements.

On this page:

- [Creating a project](#)
- [Configuring a project](#)
- [A note about project administrators](#)

Creating a project

To create a new project in JIRA:

1. Click **Projects** (in header) > **Create project**.
2. Follow the wizard to create the project.

About the project types:

- *Simple Issue Tracking*: This project provides you with a quick and easy way to get JIRA up and running for simple issue tracking. For details, see [Simple Issue Tracking project](#).
- *Project Management*: This project type creates a unique set of schemes for the project based on the default schemes.
- *Agile Software Development — Scrum (JIRA Agile required)*: This project provides you with a template for Scrum software development, including a Scrum board and Agile-specific issue types, like Story and Epic.
- *Agile Software Development — Kanban (JIRA Agile required)*: This project provides you with a template for Kanban software development, including a Kanban board and Agile-specific issue types, like Story and Epic.
- *Software Development*: This project provides you with a template to use for software development. For details, see [Software Development project](#).
- *JIRA Default Schemes*: This is the only project type that creates a project (the default JIRA project) using the *shared JIRA default schemes*. If you create a project using any other project type, the project will use project-specific schemes (i.e. issue type scheme, workflow scheme, screen scheme and issue type screen scheme).

About the project details:

- The *project key* will be used as the prefix of this project's issue keys (e.g. 'TEST-100'). Choose one that is descriptive and easy to type.
- The *project lead* is a unique project role. Choose the person who manages the project as the project lead. If there is only one user in your JIRA system, the Project Lead will default to that person and this field will not be available.

Configuring a project

To configure a project in JIRA:

1. Navigate to the administration page for the project:
 - Choose  > **Projects.**, or
 - Navigate to the desired project's summary and click the **Project Administration** button at the bottom of the project navigation sidebar.
2. Use the tabs on the left to navigate between the different project settings. Read the sections below for a description of each setting.

[Project details](#) | [Issue types](#) | [Workflows](#) | [Screens](#) | [Fields](#) | [Settings](#) | [Roles](#) | [Versions](#) | [Components](#) | [Permissio](#)

Project details

Click **Edit Project** at the top of the **Project Summary** page and edit the project details as desired. Note the following:

- Editing the project key: This is not a simple task. Read this page before you edit the project key: [Editing a Project Key](#).
- Using the Wiki Style Renderer in the project description: You can use the [Wiki Style Renderer](#) to display rich text (HTML) in your project description.
- Choosing a project avatar: If you don't want to use a project avatar, you can upload a transparent pixel.

About project categories:

The project category is not edited in the Edit Project dialog. Instead, click the link next to the **Category** field (under the project name) on the project Administration page. Categories can be viewed/created via **Administration > Projects > Project Categories**.

Why are categories useful? JIRA can search for all the issues in a particular project category (e.g. `category = "buildeng"` in an [advanced search](#)), and can display projects sorted by the project category. A JIRA project can only belong to one category. Please note that a project category is not part of a project hierarchy. Also, JIRA does not support sub-projects or parent projects.

Issue types

JIRA enables you to keep track of different types of things — bugs, tasks, helpdesk tickets, etc — by using different *issue types*. You can view the issue types that have been specified for your project, and the fields and workflow configured for each issue type. If you have the **JIRA Administrators** ([global permission](#)), you can also configure the issue types.

Click either **Issue Types** in the left menu or one of the issue types under it, e.g. **Bug, Task, Story**, etc:

- **Issue Types:** Click this to view which issue types apply to this project, (i.e. the issue type scheme). You can also view the workflow, fields and screens for the issue type in the project, but it is easier to do this by clicking one of the issue types.
If you are a JIRA administrator, click the **Actions** menu to edit the issue types in the current scheme or use a different scheme for your project.
- **One of the issue types (e.g. Bug, Task, Story):** Click this to view the workflow (**Workflow** tab)/screen (**View** tab) for the issue type in the project.
If you are a JIRA administrator, you can also configure the workflow (via the [workflow designer](#)) and the fields for the issue type ([screen designer](#)) via the tabs.

Workflows

Your JIRA issues can follow a process that mirrors your team's practices. A *workflow* defines the sequence of steps (or statuses) that an issue will follow, e.g. Open, In Progress, Resolved. You can configure how issues will transition between statuses, e.g. who can transition them, under what conditions, and which screen will be displayed for each transition.

- **Workflow Scheme** — the project's [workflow scheme](#) determines which [workflows](#) (issue state transitions) apply to issue types in this project.

Screens

JIRA allows you to display particular pieces of issue information at particular times, by defining *screens*. A screen is simply a collection of fields. You can choose which screen to display when an issue is being created, viewed, edited, or transitioned through a particular step in a workflow.

- **Screen Scheme** — the project's [screen scheme](#) determines which [screens](#) are displayed for different issue operations (view, edit, create);

OR

- **Issue Type Screen Scheme** — the project's [issue type screen scheme](#) determines which [screens](#) are displayed for different issue operations (view, edit, create), for different issue types.

Fields

JIRA enables you to define field behavior: each field can be required/optional, rich text/plain text, hidden/visible. You define this behavior by using a *field configuration*.

- **Field Configuration Scheme** — the project's [field configuration scheme](#) determines which [field configuration](#) applies to issue types in this project. (A [field configuration](#) determines each field's overall visibility, requiredness, formatting ([wiki/rich-text](#) or plain) and help-text).

Settings

- **Application Links** (Configure Project Links) — if you have linked your JIRA instance to other Atlassian applications, like Confluence, FishEye or other JIRA instances, you will be able to link this JIRA project to areas of those applications that contain information relating to your project or team. For example, Confluence spaces, FishEye repositories, JIRA projects (in another JIRA instance), etc. This allows you to take advantage of integration points between these applications. See [Linking to Another Application](#) for information about application links and project links.

Roles

Different people may play different roles in different projects — the same person may be a leader of one project but an observer of another project. JIRA enables you to allocate particular people to specific roles in your project.

- **Project Lead** — user fulfilling the role of project leader. Used as the 'Default Assignee' (see below), and potentially elsewhere in JIRA (e.g. in permission schemes, notification schemes, issue security schemes and workflows).
- **Default Assignee** — the user to whom issues in this project are initially assigned when created. Can be either the 'Project Lead' (above), or, if **Allow unassigned issues** is set to 'On' in JIRA's [general configuration](#), 'Unassigned'. There are also [default component assignees](#).
 By default, new projects also have their 'Default Assignee' set to 'Unassigned.' You can change this here if you want to set it to be a specific role, i.e. 'Project Lead.'
- **Project Roles** — members are users/groups who fulfil particular functions for this project. [Project roles](#) are used in permission schemes, notification schemes, issue security schemes and workflows.

Versions

If you are using JIRA to manage the development of a product, you may want to define different *versions* to help you track which issues relate to different releases of your product (e.g. 1.0, 1.1, 1.2, 2.0 beta, 2.0). JIRA can help you manage, release and archive your versions. Versions can also have a Release Date, and will automatically be highlighted as "overdue" if the version is unreleased when this date passes.

- **Versions** — versions defined in the project. See the [version management](#) page for details.

Components

You may want to define various *components* to categorise and manage different issues. For a software development project, for example, you might define components called "Database", "Usability", "Documentation" (note that issues can belong to more than one component). You can choose a Default Assignee for each component, which is useful if you have different people leading different sub-teams in your project.

- **Components** — logical groups that this project's issues can belong to. See the [component management](#) page for details.

Permissions

JIRA allows you to control who can access your project, and exactly what they can do (e.g. "Work on Issues", "Comment on Issues", "Assign Issues"), by using *project permissions*. You can also control access to individual

issues by using *security levels*. You can choose to grant access to specific users, or groups, or roles (note that roles are often the easiest to manage).

- **Permission Scheme** — the project's [permission scheme](#) determines who has permission to view or change issues in this project.
- **Issue Security Scheme** — the project's [issue security scheme](#) determines what visibility levels issues in this project can have (see [issue-level security](#)).

Notifications

JIRA can notify the appropriate people when a particular event occurs in your project (e.g. "Issue Created", "Issue Resolved"). You can choose specific people, or groups, or roles to receive *email notifications* when different events occur. (Note that roles are often the easiest to manage.)

- **Notification Scheme** — the project's [notification scheme](#) determines who receives email notifications of changes to issues in this project.
- **Email** — specifies the 'From' address for emails sent from this project. Only available if an [SMTP email server](#) has been configured in JIRA.

 Please note, the **Default Notification Scheme** (shipped with JIRA) is associated with all new projects by default. This means that if you have an [outgoing \(SMTP\) mail server](#) set up, that email notifications will be sent as soon as there is any activity (e.g. issues created) in the new project.

Development tools

The Development tools section gives you an overview of the development tools that are connected and which users can use the integration features between them:

- **View permission** - This section lists which users can see the development tools integration features (like the **Create Branch** link) on the view issue screen, as well as other development-related information, like commits, reviews and build information. This ability is controlled by the "View Development Tools" project permission.
- **Applications** - This section shows which development tools are connected to JIRA via application links and are eligible to use the development tool features in JIRA.

A note about project administrators

A project administrator in JIRA is someone who has the project-specific **Administer Projects** [project permission](#), but not necessarily the **JIRA Administrator** [global permission](#).

Without the **JIRA Administrator** [global permission](#), however, project administrators can do the following:

- Edit the project name
- Edit the project description
- Edit the project avatar image
- Edit the project URL
- Edit the project lead
- Edit [project role membership](#)
- Define [project components](#)
- Define [project versions](#)
- View, but not select nor edit the project's schemes (notification scheme, permission scheme, etc)

Changing the project category of a JIRA project requires **JIRA Administrator** [global permission](#).