



# Jira Administration Part 1

## Data Center and Server

Welcome to the Jira Administration Part 1 for Data Center and Server course.

1

# Course Overview

21 Jan 2021 – Data Center and Server



Our first module provides an overview of the course content.

# What will you learn?



- Configure system settings
- Set up users and groups
- Configure global permissions
- Create and configure projects
- Create issue types, update workflows, and edit fields and screens
- Configure project permissions and roles



What will you learn in this course? You'll learn the basics of Jira Administration. We'll look at configuring system settings, setting up users and groups, and we'll be looking at global permissions. We'll take a look at configuring projects, which includes creating and managing issue types, fields and screens, and workflows. Finally, we'll look at project-level permissions and the advantages of using project roles.

## To succeed here, you need to have



- Basic Jira knowledge
- A basic understanding of projects, issues, issue types, and workflow
- Downloaded lab courseware
- Tested the lab environment



Before taking this course, you should know the basics of using the Jira application(s) you have – Jira Software, Jira Service Management, and/or Jira Core. You should also have downloaded the lab courseware and tested the lab environment. We'll look at setting up a lab environment for in a moment.



## Course Overview

Configuring Your System

Setting-Up Users & Groups

Configuring Global Permissions

Creating & Configuring Projects

Configuring Project Roles & Permissions



Here are the six modules that we'll cover.

# Live teach schedule



1	Course Overview	15 minutes
2	Configuring Your System	1 hour 25 minutes
	Break	10 minutes
3	Setting-Up Users & Groups	1 hour 10 minutes
4	Configuring Global Permissions	50 minutes
	Lunch	40 minutes
5	Creating & Configuring Projects	2 hours 5 minutes
	Break	10 minutes
6	Configuring Project Roles & Permissions	1 hour 15 minutes
	Total	8 hours



And here is a schedule for the course.

# Try it

## Starting Your Lab Environment



- Use the link provided to access your lab VM
- Start your Jira instance



In the first lab for this course, we set up a lab environment.

The lab environment consists of a Jira instance that has been set up for you to use to complete the hands-on exercises for this course. You'll receive a link that you can use to access your lab VM. After you click the link provided to access your lab VM, it takes around ten minutes for Jira to initialize. At that point, you can click the link to start using the lab environment. We'll take five minutes to go through the process of initializing the lab environment.

## 2 Configuring Your System



In this module, we'll look at configuring your Jira instance.

# What will you learn?



- Outline your mission as a Jira administrator
- Identify the three applications that make up the Jira platform
- Integrate Confluence with Jira and install an app
- Brand your look and feel and customize the system dashboard
- Locate the Jira application log and use support tools for troubleshooting
- Configure auditing to match your company's retention policies
- List the steps to enable public access and customize time tracking



Here's what you'll learn in this module.

We'll start by considering your mission as a Jira administrator. Then we'll look at the three Jira applications that make up the Jira platform.

We'll look at integrating Confluence with Jira and installing an app. Another general system configuration that we'll look at is changing the default look and feel of Jira so that it matches your corporate branding, and we'll customize the system dashboard.

When managing your Jira instance, you'll need to know how to locate the Jira application log and how to use the support tools for troubleshooting, so we'll cover that, too. We'll look at auditing in Jira, and finally, we'll cover customizing time tracking and enabling public access for Jira.



# Your Mission as a Jira Administrator

Be the Jira Evangelist

Make Jira easy to use

Ensure your company gets value from Jira

Build it, ship it, then get user feedback and update Jira

Jira needs to constantly evolve as it adapts to your business!



Let's start with your mission; that's your mission as a Jira Administrator. As a Jira Administrator, you own the experience of using Jira, which means that you are the Jira Evangelist.

It's your mission to make Jira easy to use and to make sure that your company gets value from Jira.

As an administrator, the expectation is that you configure the Jira instance based on end-user and stakeholder requirements. You release a configured Jira to users, and they start using it. Then it's up to you to get user feedback and update Jira accordingly. You must take the time to listen to the users and stakeholders, understand their pain points so that you can learn from their experience.

For example, if users complain that issues are complicated by too many fields or that they don't understand why they should use a field, then you should make changes. Use a Jira project, the Jira Issue Collector, or ServiceDesk if you have access to the application to gather user feedback and change requests. Jira configurations need to evolve as they adapt to your business continuously.

# What is the Jira Platform?



◆ Jira Software

Software teams

⚡ Jira Service Management

IT teams

↑ Jira Core

Business teams



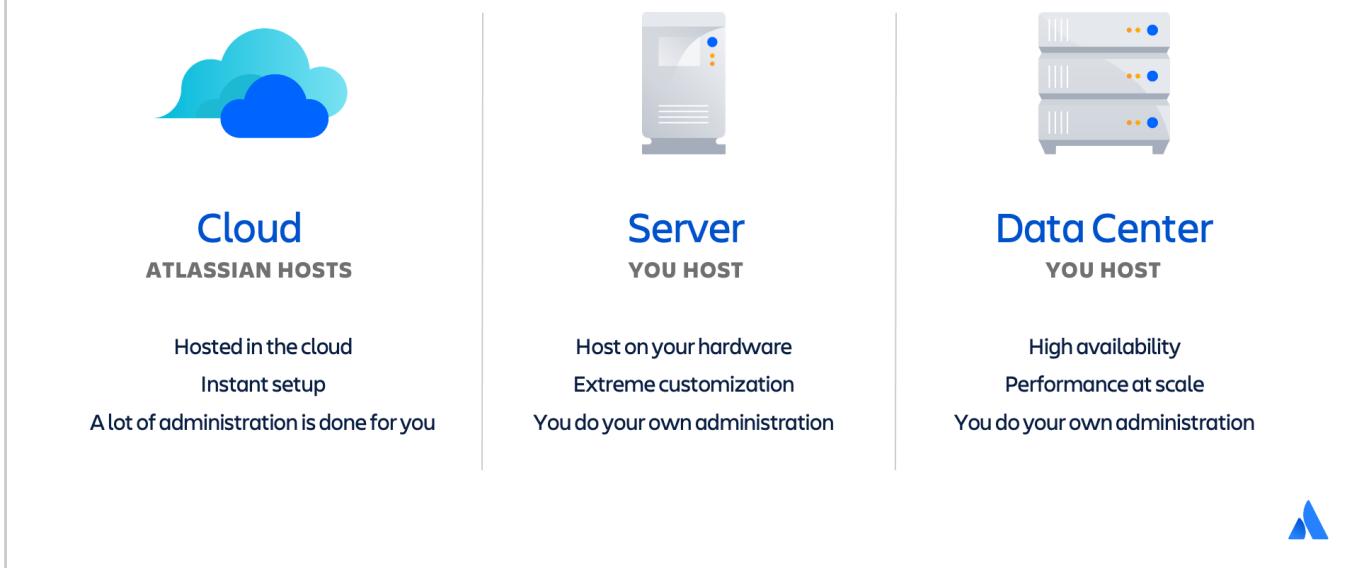
Jira is an issue tracking system that is frequently used within the software development field. However, Jira is flexible enough to support other areas of focus outside of software development. Many non-software development project teams use Jira for tracking issues as well as Finance teams and more.

The Jira Platform consists of three products. These products match the three core audiences. Each Jira product comes complete with a specific set of features and functions, which tailor the experience delivered to Jira users.

Jira Software is designed for agile teams whose focus is software development. It enables these teams to plan projects, manage dependencies, and track team progress. Jira Service Management is designed for IT and service teams to implement service desks quickly and to manage incidents, problems, changes, and service requests.

Jira Core is designed for business teams, whose goals are to keep their teams organized by planning, tracking, and reporting on their business projects. It's a lightweight version of Jira Software, stripping away all unnecessary features for Business teams such as sprint planning. If you have Jira Software or Jira Service Management, you don't need a separate subscription for Jira Core because you automatically get all the functionality of Jira Core.

# Jira deployment options



Jira has three deployment options; Cloud, Server and Data Center. This course focuses on Jira Server.

Cloud is aimed at businesses that want minimal overhead in system administration. They do not want to maintain servers, software upgrades, and other things that an IT system administrator, server administrators and a host of others must take care of. There is a System Administrator in the Cloud but it's only accessible by Atlassian support. All functions performed by the Jira system administrator in Jira server applications are restricted in Jira Cloud applications. This option has a monthly or annual subscription. Server runs on your hardware and can sit behind your firewall. Jira Server is aimed at large businesses and teams that plan to integrate Jira with their own internal systems. Also some businesses choose not to use Cloud because they do not want any part of their data in the cloud and sometimes some organizations need full control over everything. Jira Server has a perpetual license.

Data Center is designed for high availability and performance at scale when hosting Atlassian applications in your own data center. It's for organizations that have the need for scale, specifically when they have many concurrent users. It has an annual term license.

Supplemental information:

For the pros and cons of Cloud vs. Server, see

<https://confluence.atlassian.com/display/CLOUDKB/Pros+and+Cons+of+Cloud+vs.+Server>.

# Versions & Licenses

If the license expires, you can't access tech support or application updates

Uninstall the application

Jira Software 8.5.1	
Unlimited users	(1 used)
Maintenance expires	30/Dec/22
Support entitlement number (SEN)	SEN-[REDACTED]
License type	Developer
Organisation name	Atlassian
License key	AAABfQ0ODAoPeNp9k...  
Uninstall	

# users licensed and used

Update your license



If you exceed the licensed user count, your users won't be able to create issues.



On the Versions & licenses page you can:

- See what Jira applications are installed
- And you can change the type of license that is used
- You can upgrade your user tier to accommodate new users
- And you can add a new license when your old license has expired
- You can add a new license for a newly installed application
- And you can view details of your licenses. This includes the licensed user count for your Jira instance, maintenance expiry, license type, and so on.

You may want to purchase a full license when your evaluation license expires.

With licensing, you should be aware of what happens if you exceed your user count. If you exceed your user count, you need to either upgrade to a larger license or reduce your existing user count. If you're near the maximum users for your license, you'll see a banner prompting you to upgrade. You can click the Upgrade button in the banner to redirect to the Atlassian order form.

Every user in Jira has access to an application based on their membership of groups.

We'll cover groups and application access in more detail in the next module. The important thing to know is that a user may have access to all the applications, or only one application. If a user has access to an application, they count as one licensed user for that application. E.g., if a user belongs to a group for Jira Software and a group for Jira Service Management, they count as a licensed user for both Jira Software and Jira Service

Management.



## Top tasks to configure Jira

- Integrate other products
- Install apps

Integrating products and apps

- Brand your look & feel
- Set the default language

User experience

- Customize time tracking
- Configure auditing
- Find the Jira log and configure auditing
- Enable public access and CAPTCHA
- Permissions & project configuration...

Other configurations

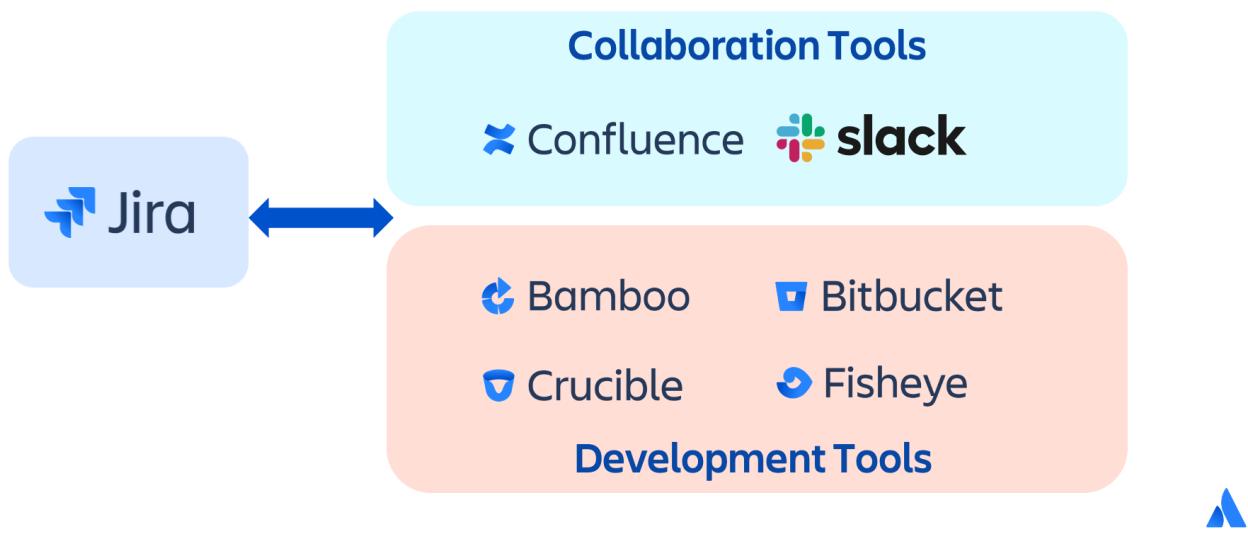
Covered in later modules



There are many ways to configure Jira. These are the top tasks and are the ones covered in the rest of this module.

Global permissions and project configuration are also very important tasks to configure your Jira. These topics will be covered in later modules.

# Integrating Other Atlassian Products



You can easily integrate other Atlassian products with Jira via the Applications administration section under INTEGRATIONS. You can also integrate with collaboration tools such as Confluence and Slack. Confluence gives your teams the ability to share, discuss, and work with Jira application issues in Confluence, and create knowledge articles for your service desk customers. Integrating Jira applications and HipChat gives you and your teams the following collaboration power. You can:

- Get notifications in your Slack channels when a customer updates a service desk request, or a developer comments on an issue.
- Create a dedicated Slack channel from the issue you're working on and want to discuss it with your team.
- Create issues in Jira from Slack
- Preview Jira issues and service desk requests directly in Slack when someone on your team mentions them.

See <https://community.atlassian.com/t5/Jira-articles/Connect-Jira-and-Confluence-Server-to-Slack-Today/ba-p/1008234>

You can also integrate with Jira with development tools such as Bamboo, Bitbucket, FishEye/Crucible, and Github. Connecting Jira Software to compatible development tools provides your team with a range of functionality and information related to your development work.

We'll cover integrating Jira with Confluence, which we'll look at in the next slide.

# Integrating Jira & Confluence



Lets users access Confluence quickly

Add your own links, reorder links, etc.

Control appears when you add a link



Confluence and Jira are like bacon and eggs or coffee and cake. Separately, they're great, but together, they're amazing! Linking these applications together allows your users to display and create Jira issues and more from Confluence. It also allows Jira Software users to link and create Confluence pages from epics and sprints. Typically Jira and Confluence are installed together. To link the applications, you just need to know the Confluence URL. Note that the URL may not include a port number if, for example, you use reverse proxies.

Once you link the applications, a link to Confluence will automatically be added to the Application Navigator as you see here.

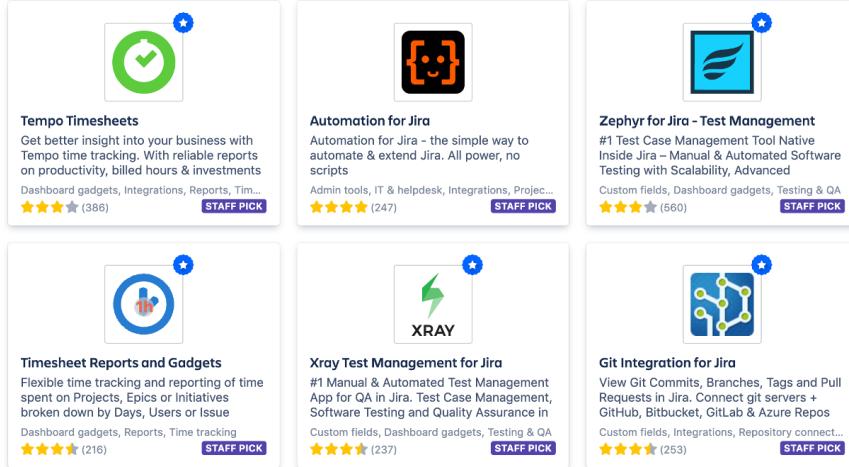
Linking two applications allows you to share information and access one application's functions and resources from within the other.

The application navigator is the icon (a grouping of nine squares) in the top left of the Jira header that displays a menu of links to other applications. It is only displayed to users if there is more than one link. You can add custom links in the application navigator, to make it easier for users to navigate to frequently used information. By adding a link to Confluence here, users can quickly and easily move between the applications.

You can customize the links that appear in the application navigator, as well as making certain links only visible for specific users.

# Installing Apps

 **ATLASSIAN** Marketplace



The screenshot shows a grid of six app cards from the Atlassian Marketplace:

- Tempo Timesheets**: Get better insight into your business with Tempo time tracking. With reliable reports on productivity, billed hours & investments. Dashboard gadgets, Integrations, Reports, Tim... **STAFF PICK**
- Automation for Jira**: Automation for Jira – the simple way to automate & extend Jira. All power, no scripts Admin tools, IT & helpdesk, Integrations, Projec... **STAFF PICK**
- Zephyr for Jira - Test Management**: #1 Test Case Management Tool Native Inside Jira – Manual & Automated Software Testing with Scalability, Advanced Custom fields, Dashboard gadgets, Testing & QA **STAFF PICK**
- Timesheet Reports and Gadgets**: Flexible time tracking and reporting of time spent on Projects, Epics or Initiatives broken down by Days, Users or Issue Dashboard gadgets, Reports, Time tracking **STAFF PICK**
- Xray Test Management for Jira**: #1 Manual & Automated Test Management App for QA in Jira. Test Case Management, Software Testing and Quality Assurance in Custom fields, Dashboard gadgets, Testing & QA **STAFF PICK**
- Git Integration for Jira**: View Git Commits, Branches, Tags and Pull Requests in Jira. Connect git servers + GitHub, Bitbucket, GitLab & Azure Repos Custom fields, Integrations, Repository connect... **STAFF PICK**



Add functionality but ensure there's a business requirement

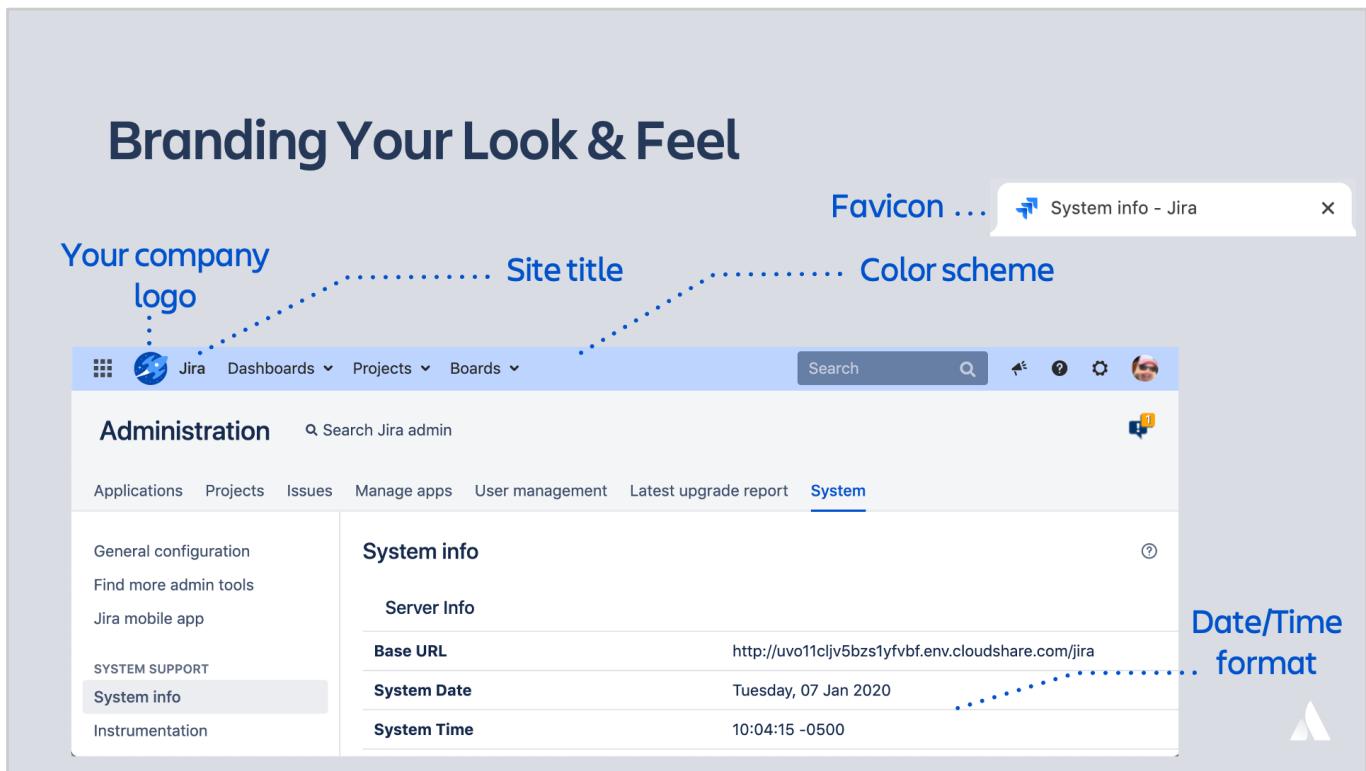


An app (which was previously known as an add-on, and before that a plugin) is an installable component that supplements or enhances the functionality of Jira in some way. For example, the Jira Calendar Plugin is an app that shows the due dates for issues and versions in calendar format. Other apps are available for connecting to Bamboo, developing for Jira, and accessing Atlassian support from Jira.

Jira comes with many pre-installed apps (these are called system apps). You can install more apps, either by downloading an app from the Atlassian Marketplace or by uploading an app from your file system. This means that you can install apps that you've developed yourself. The apps available in the Atlassian Marketplace are from Atlassian and third-party vendors. There are over 2,000 apps available.

Governance is important. If you install an app, users will start using it, and they'll get used to the extra functionality. If you remove it, you may have unhappy users. For example, you may install an app for extended JQL functionality. Then you disable it because the system has grown, more people are using it, and you notice that there's a performance hit. During this time, however, users have gotten used to the JQL extensions, and now all custom JQL will be invalid. Filters using that JQL will no longer work. Users will not be frustrated. The important point is that before you install an app, ensure there's a business requirement for it. Don't just install any app. Do your due diligence before installing it. The terms 'app' and 'plugin' and 'add-on' are often used interchangeably, but they are all essentially the same thing.

# Branding Your Look & Feel



You configure your branding on the Look and Feel page. For example, you can upload or link to your company's logo. The color scheme will be updated automatically to match your logo, or you can set the colors individually.

The Site Title is the name of the instance. It's the text that appears to the right of the logo if enabled. Here, it's Jira, but you can change that to anything that works for your organization by updating the title in site settings.

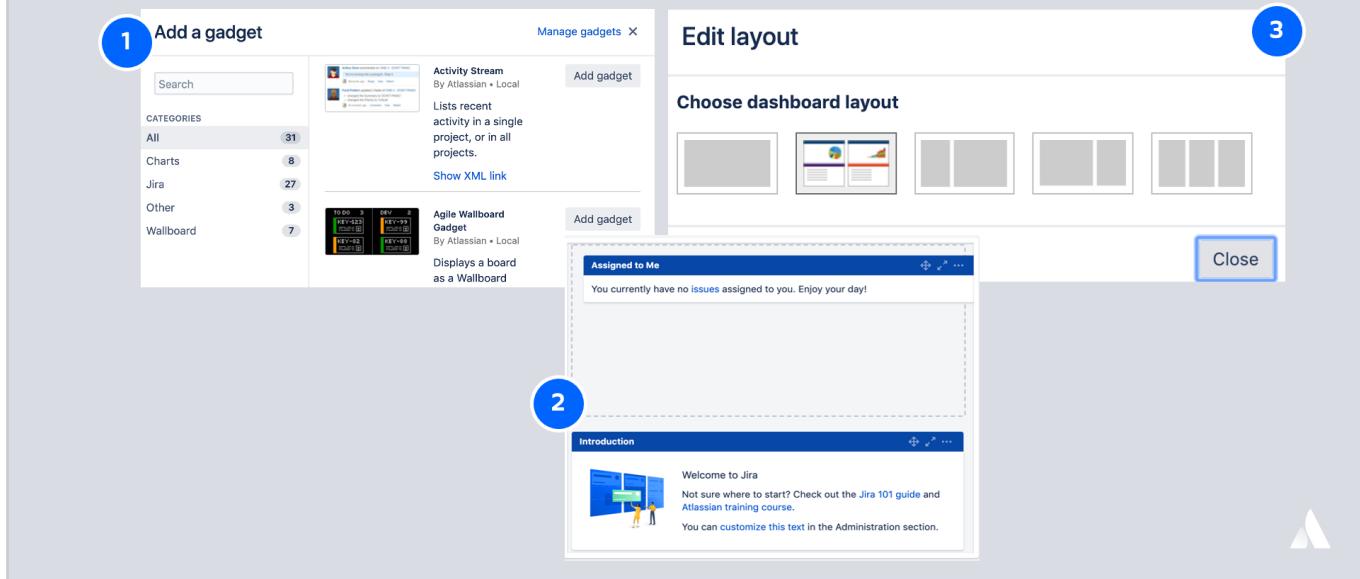
You can also upload or link to your own Favicon, which typically appears to the left of your browser's URL field and on browser tabs displaying a page on your Jira site. It must be a png.

The Date/Time format allows you to customize the way dates and times are presented to users in Jira.

Here you see a Jira instance that has been customized with a favicon and a new logo (thus changing the color scheme). The Site Title is also enabled.

See [design.atlassian.com](https://design.atlassian.com) for principles, guidelines, and assets for designing and building awesome Atlassian experiences.

# Customizing the System Dashboard



A further area that you should configure to meet your organizational needs is the system dashboard. The system dashboard is the screen that all Jira users see the first time they log in. Any users who have not added any dashboard pages as favorites also see the system dashboard. The system dashboard can contain the same gadgets that are available to any dashboard and configurations work in the same way.

Jira administrators can configure the system dashboard by:

1. Adding gadgets to the dashboard. Gadgets are the information boxes on the dashboard. Jira comes pre-configured with a set of standard dashboard gadgets.
2. Re-ordering gadgets on the dashboard or switching them between the left and right columns.
3. Configuring the layout of the dashboard (such as changing the number of columns). The gadgets on the system dashboard can be configured. All changes made to the system dashboard updates the system dashboards for all users. Note that gadgets do not display any issues to users if they do not have permission to view them. Also, if any gadgets have been populated using a filter, then the filter must be shared correctly; otherwise, users cannot see the gadget content.



## Are you getting it?

What will happen if you exceed the number of licensed users?

- a. Only the Administrator will be able to log into Jira, other users will be locked out.
- b. Only the licensed number of users will be able to log into Jira.
- c. Users won't be able to create issues/tasks.



Answer is on the next slide.

## Did you get it?



What will happen if you exceed the number of licensed users?

- a. Only the Administrator will be able to log into Jira, other users will be locked out.
- b. Only the licensed number of users will be able to log into Jira.
- c. Users won't be able to create issues/tasks.



Answer: c. Users won't be able to create issues/tasks if you exceed the number of licensed users.

# Are you getting it?



Which statement is true?

- a. A limited subset of gadgets is available for use on the system dashboard.
- b. Any user can view the system dashboard.
- c. Any user can update the system dashboard



Answer is on the next slide.

## Did you get it?



Which statement is true?

- a. A limited subset of gadgets is available for use on the system dashboard.
- b. Any user can view the system dashboard.
- c. Any user can update the system dashboard



Answer: b



## Takeaways

- Don't exceed the licensed user count or your users won't be able to create issues
- Create application links for frequently used applications
- Install apps to enhance the functionality of Jira but ensure there's a business requirement for it first



Here are some of the headlines to take away from this module.

You should make sure that you don't exceed the licensed user count otherwise the system will not allow users to create issues.

You can customize the links that appear in the application navigator in Jira, as well as making certain links only visible for specific users. This lets users quickly access other frequently used applications.

With apps, governance is important. If you install an add on, users start using it and get used to the extra functionality. Then, if you remove it, you may have unhappy users. The important point is that before you install an app, ensure there's a business requirement for it, don't just install any app. Do your due diligence before installing apps.

**“Our company has quickly grown over time. Now we have different teams, distributed all across different planets working on various projects.”**



## Teams in Space

### Your Task Today

Setup up and configure the Jira instance so teams can start using it



Teams in Space is a fictional company that specializes in space travel for teams. This is the case study we use throughout the course. The company is distributed across the universe.

The company has just installed Jira to allow their distributed teams to run more efficiently, where tasks are visible, shared, and easy to access.

The scenario is that you are the Jira administrator for Teams in Space. The Jira applications (Software, Service Management, and Core) have just been installed. Now you need to set up and configure the instance so Teams in Space's teams can start using it. You have one day, so let's get started.

## See how it's done



- How to connect to the lab environment
- Download the lab resource file
- Log in to Jira
- Versions & licenses administration page



In this demo we'll look at how to:

- connect to the lab environment
- Download the lab resource file
- Log in to Jira
- And we'll look at the versions & licenses administration page

# Try it

## Lab 2 - Configuring Your System



- Exercise 1 – Verifying Applications & Configuring Integrations
- Exercise 2 – Configuring the User Interface



Next, we have a lab, which contains hands-on exercises relating to configuring Jira. The Lab Workbook contains both high-level Instructions and detailed instructions. Use the high-level instructions for less guidance if you feel like a challenge. Use the detailed Instructions for step-by-step instructions.

# Setting the Default Language



Most user-visible pages in JIRA are internationalized. When JIRA is first installed, you can select from several default languages, including English (UK or US), French, Japanese, Spanish (Spain), and German.

You can also get plug-ins for Chinese, Czech, Danish, Italian, Norwegian, Polish, Portuguese (Brazilian), Russian, and Slovak with more in development.

You can change the default language for Jira on the General configuration administration page (under System).

Supplemental information:

For a list of all the languages available, see <https://confluence.atlassian.com/translations>

# Customizing Time Tracking

Setting	Default Setting
The number of working hours per day	8
The number of working hours per week	5
Format of time estimates	Pretty e.g. 4 days, 4 hours, 30 minutes
Unit for time tracking	Minute
Copying of comments to work description	Enabled



Jira's time tracking feature enables users to record the time they spend working on issues. An Administrator can configure time tracking settings and can disable time tracking altogether.

Configure time tracking settings by editing the following fields:

- 'Hours per day' – enter a suitable value (e.g. 8). You can enter fractions if you wish.
- 'Days per week' – enter a suitable value (e.g. 5). You can enter fractions if you wish.
- 'Time format' – select pretty/days/hours. This will determine the format of the 'Time Spent' field when an issue is displayed.
- 'Default Unit' – select minutes/hours/days/weeks. This will be applied whenever your users log work on an issue without specifying a unit.
- 'Legacy Mode' – select this checkbox if you prefer to use Jira's time tracking features as they operated prior to Jira version 4.2.
- 'Copy Comment To Work Description' – select this checkbox to ensure that any content entered into a Comment field while logging work as part of an issue operation, is also copied across to the Work Description. When 'Copy'
- If 'Comment To Work Description' is enabled, your user's work log entries will be visible only to members of the project role or group selected in the padlock icon dropdown on their issue operation screen. If 'Copy Comment To Work Description' is disabled, your user's work log entries will be visible to anyone by default.

# Logging

<Jira Home>/log/atlassian-jira.log

Logging Levels:

DEBUG  
INFO  
WARN  
ERROR  
FATAL



```
JIRA starting...
-----
2017-10-22 10:22:16,765 JIRA-Bootstrap INFO [c.a.jira.startup.JiraStartupLogger]

Environment
-----
JIRA Build : 7.5.0#75005~sha1:fd8c849d4e278dd#bbaccc61e707a716ad697024
JIRA Installation Type : True Sep 05 00:00:00 EDE 2017
Application Server : Standalone
Java Version : 1.8.0_144 - Oracle Corporation
System Working Directory : /opt/jira/jira
Maximum Allowable Memory : 683MB
Total Memory : 455MB
Used Memory : 412MB
Memory Pool: Old Cache : 43MB
Memory Pool: New Cache : Code Cache: init = 2555904(2496K) used = 8301568(8107K) committed
Memory Pool: Compressed Class Space : Metaspace: init = 0(0K) used = 21655616(21149K) committed
Memory Pool: PG Eden Space : Compressed Class Space: init = 0(0K) used = 2537664(2478KK)
Memory Pool: PG Survivor Space: init = 100663296(98304K) used = 21936872(22
Memory Pool: PG Old Gen: init = 16777216(16384K) used = 0(0K) committed
Memory Pool: PG Old Gen: init = 258405456(262144K) used = 23409376(228
Memory Pool: PG Old Gen: init = 536875932(501288K)
Memory Pool: PG Old Gen: init = 536875932(501288K)

Atlassian - 2017-10-22 10:22:16,765 JIRA-Bootstrap INFO [c.a.jira.startup.JiraStartupLogger]
```



The application log file, `atlassian-jira.log` is written to the `log` subdirectory of your Jira application home directory. Find the exact location in `System > System info, File Paths`. Security-related information e.g. login, logout, session creation/destruction, security denials is stored in `atlassian-jira-security.log`.

There are five logging levels available in log4j: 'DEBUG', 'INFO', 'WARN', 'ERROR' and 'FATAL'. Each logging level provides more logging information than the level before it. 'DEBUG' provides the most verbose logging and 'FATAL' provides the least verbose logging. The default level is `WARN`, meaning warnings and errors are displayed.

Sometimes it is useful to adjust this level to see more detail. Please be aware: the 'DEBUG' setting may cause user passwords to be logged.

The default logging levels can be changed either:

- Temporarily - your change will not persist after you next restart Jira, or
- Permanently - your change will persist, even after you restart Jira.

For example, when troubleshooting, you might temporarily change the logging level from 'WARNING' to 'INFO' to get a more detailed error message or a stack trace. If you're not sure which logging categories to adjust, the most helpful information generally comes from the `log4j.rootLogger` category and the `log4j<category>.com.atlassian` categories.

You can also set retention for the log files themselves. For more information on logging, including editing the logging settings in `log4j.properties`, see

<https://confluence.atlassian.com/display/ADMINJIRASERVER085/Logging+and+profiling>.

# Troubleshooting



- Talk to an experienced System Administrator



- Use Support Tools to create a zip to send to Atlassian Support



We won't go into troubleshooting practices here as it's beyond the scope of this course. However, two recommendations we will make at this level are to talk to an experienced System Administrator at your organization and also to use the Support Tools within Jira to create a Support zip file to send to Atlassian Support.

The Support Tools plugin has been developed by Atlassian Support to help you:

- Keep an eye on the health of your instance, and alert you problems with your configuration
- Log a Support Request
- Create a Support Zip, which contains configuration and diagnostic information for Atlassian Support. This file is saved in <JIRA\_HOME>/export/Jira\_support\_<timestamp>.zip.
- Scan your instance logs against our knowledge, to help you diagnose and troubleshoot known issues
- Schedule automatic log scans and summaries

Supplemental information:

More information on the Support Tools can be found at

<https://confluence.atlassian.com/display/Support/Support+Tools+Plugin>.

# Auditing

- Tracks key activities
- Records configuration changes
- Handy tool to diagnose problems
- Follow your company's retention policies!



Category	Events
Auditing	auditing enabled, auditing disabled
LDAP synchronization	LDAP synchronization
User management	user added, user removed, user changed
Group management	group added, group removed, user added to group, user removed from group
Project changes	project created, project removed, project updated
Permission changes	scheme created, scheme copied, scheme removed, scheme edited, scheme assigned to a project, scheme unassigned from a project, permission added to scheme, permission removed from scheme, global permission added to a group, global permission removed from a group
Workflow changes	scheme created, scheme copied, scheme removed, scheme edited, scheme assigned to a project, scheme unassigned from a project, workflow created, workflow copied, workflow removed, workflow renamed, workflow draft published
Notification changes	scheme created, scheme copied, scheme removed, scheme edited, scheme added to project, scheme removed from project, notification added to scheme, notification removed from scheme
Screen changes	scheme created, scheme copied, scheme removed, scheme edited, scheme added to project, scheme removed from project, screen added to scheme, screen removed from scheme, screen field configuration changed
Custom field changes	custom field created, custom field updated, custom field removed, scheme added to project, scheme removed from project
Workflow changes	Status added, status updated, status deleted, transition added, transition updated, transition deleted



You can access the Jira audit log via System > Audit Log. It's a handy tool to help you diagnose problems. It tracks records most, but not all configuration changes that can impact users and projects.

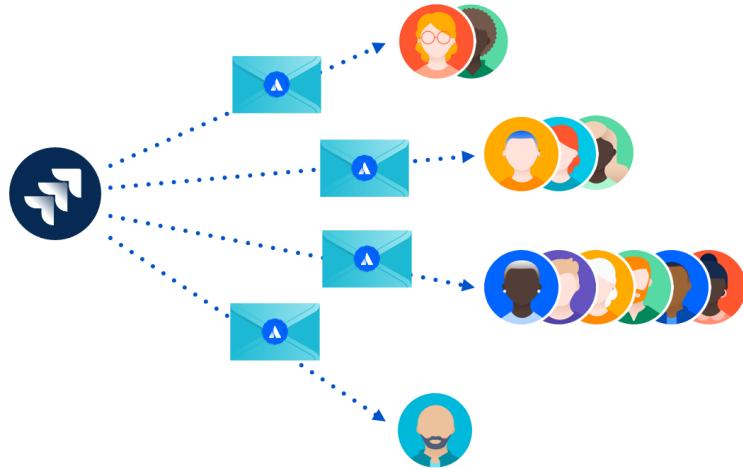
You can change the retention period from the default of unlimited. If you modify the audit log retention period, ensure you follow your company's retention policies!

You can also configure the audit log not to include events triggered by your external repository.

Supplemental information:

See <https://confluence.atlassian.com/display/AdminJiraServer085/Auditing+in+Jira+applications>.

# Enabling Email Notification



For Jira to send email notifications, you need to configure an SMTP server in Jira



Notifications allow you to send emails to users when particular events occur. For example, when you create a new user, you can select to notify them by email or even invite them by email. You can also set up notifications for other events such as issue created, updated, resolved, and so on.

Take care when enabling notifications as your users may feel inundated if they receive too many Jira notifications. We'll discuss how to edit notifications later in the course. For Jira to send email notifications about various events, you need to configure an SMTP server in Jira. Set up the outgoing server through System > Outgoing Mail.

Supplemental information:

See

<https://confluence.atlassian.com/display/AdminJiraServer085/Creating+a+notification+scheme> for a list of all the events that can trigger notifications.

# Private Access vs. Public Access

Only administrators/user directory managers can create new users



Only our employees will be using Jira and we want their accounts created for them...

Visitors can immediately create their own accounts



We want to use Jira as a support system and have a very large number of potential users. Only some users will need to log support tickets.



Jira can operate in two modes:

**Private.** In private mode, only administrators can create new users if the Jira Internal Directory is used. However, if you use an external user repository such as LDAP and Jira set to read-only, then users will be created in the external user directory by the user directory manager (which will probably be a different person than the Jira administrator).

**Public.** In public mode, any user can sign up and create an account. If these users create accounts that give them access to Jira Core or Jira Software, these accounts will consume a license for these applications. Think carefully before enabling public signup. Consider your license limits. New users are automatically added to the default application group, so be aware of what they'll be able to do. For security reasons, even if you enable signup, it is still necessary for users to have the appropriate project permissions before they can see or create issues. We'll cover project permissions later in the course.

It's also possible to allow people anonymous access to a project where they could browse and search for issues or even create issues without logging in. That is a project permission issue. We won't cover that here. But generally, we recommend that if you allow anonymous access, you limit it to read-only activities such as browsing. If you allow users who haven't logged in to create issues, the changes don't get logged because Jira doesn't know who the user is.

## Enabling CAPTCHA

- Self-signup systems are targets for 'bots'
- CAPTCHA makes them prove they're human



Sign up

---

Email\*

Full name\*

Username\*

Password\*  ?

Please enter the word as shown below

---



There are risks in allowing people to sign up for accounts in a publicly available system. If you allow access to Jira from outside your organization's firewall, it will most likely become a target for automated spam systems (or 'bots') to subvert it for "Spam" or worse. In this case, you may want to enable CAPTCHA. When CAPTCHA is enabled, visitors have to recognize a distorted picture of a word as you can see in the screenshot, and must type the word into a text field. This is easy for humans to do, but very difficult for computers.

# Are you getting it?



Where is the Jira application log located?

- a. In the Jira Home directory.
- b. In the Analytics area of Jira Administration.
- c. In the System info area of Jira Administration.



Answer is on the next slide.

## Did you get it?



Where is the Jira application log located?

- ✓ a. In the Jira Home directory.
- b. In the Analytics area of Jira Administration.
- c. In the System info area of Jira Administration.



Answer: a



## Takeaways

- View the application and audit logs when there's a problem
- If you need more help, talk to an experienced System Administrator or use the Support Tools to create a zip to send to Atlassian Support
- Follow your company's retention policies for the audit log
- If you enable public signup, also enable CAPTCHA to avoid 'bots



Here are some of the headlines to take away from this module.

The audit log in Jira will help administrators to understand where and when configuration changes were made, and who made them.

If you are experiencing problems with your Jira instance, you should first check with an experienced system administrator at your organization, and you have the option to use the support tools to create a zip and send it to Atlassian support.

If you make any changes to the audit log's settings, you should make sure that you follow your company's retention policies when doing so.

And finally, if you enable public signup, you should also make sure that you enable CAPTCHA to avoid 'bots.

# Try it

## Lab 2 - Configuring Your System



- Exercise 3 – Log Files, Auditing & Support Tools
- Optional Exercise 4 – Exploring Configuration Settings



Next, we have a lab that looks at configuring your system.

# 3 Setting Up Users & Groups



In the next module, we'll look at setting up users and groups.

# What will you learn?

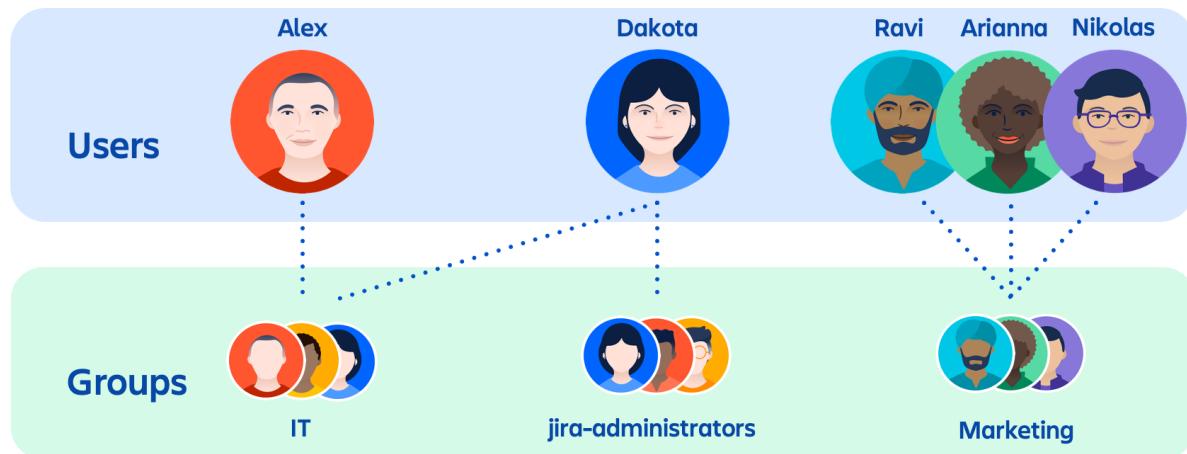


- Control application access with groups
- Create users
- Troubleshoot access issues



We'll look at controlling access with groups, creating users, and troubleshooting access issues.

# Users & Groups



Users are the people who access Jira applications. You need to add users to Jira. The one exception is the Admin user who is created when you install Jira. In this example, Alex, Dakota, Ravi, Arianna, and Nikolas are all users.

A Jira group is a convenient way to manage a collection of users. Users can belong to many groups. In this example, Dakota belongs to both the IT and jira-administrators groups. Alex belongs to the IT group and Ravi, Arianna, and Nikolas belong to the Marketing group.

Groups are:

- Used throughout Jira to manage users (for example to provide access to applications)
- Granted global permissions (this is discussed in next module)
- Used in project permission schemes, email notification schemes, issue security levels, and workflow conditions.

Groups can also be given access to issue filters and dashboards, and belong to project roles. We will cover these areas later in the course.

# Creating Usernames

- Consistent naming e.g. jsmith, hjennings...
- Don't use email address
- Must be unique

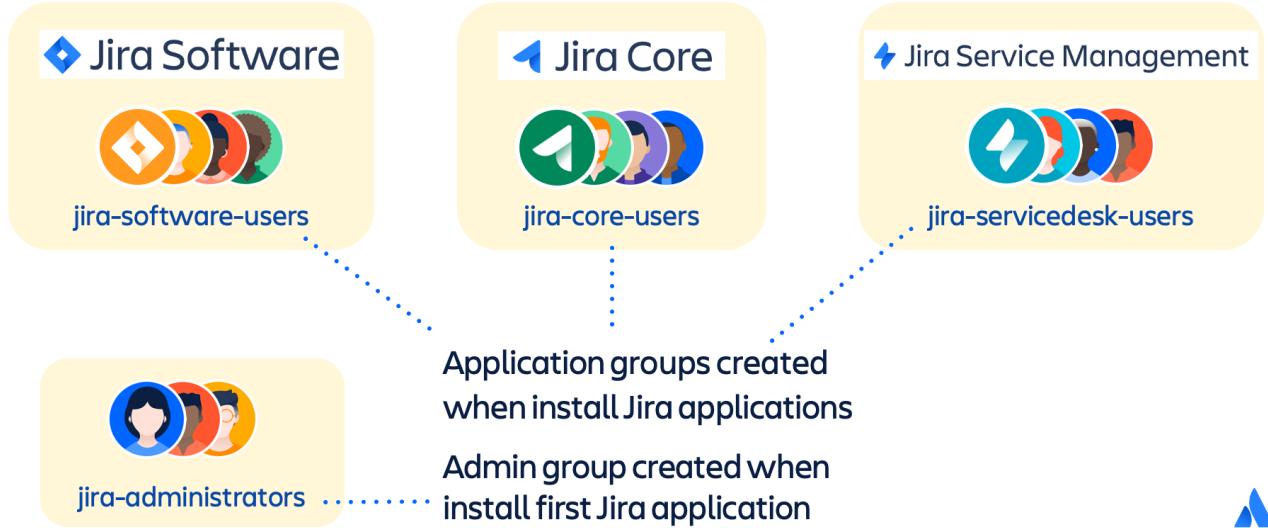


When creating users, use a consistent naming convention. For example:

- <first letter of first name ><last name>
- jsmith, agrant, hjennings ...

Usernames must be unique. Do not include email-addresses in user names.

# Default Groups for each Jira Application



Each Jira application has a default group associated with it. The default application groups are:

- `jira-software-users` for Jira Software
- `jira-core-users` for Jira Core
- `jira-servicedesk-users` for Jira Service Management

There is also a `jira-administrators` group for the user(s) who perform global administration of your Jira instance.

When you install your first Jira application and license it, Jira will create two user groups, and add the Jira administrator, to both of them – `jira-administrators` group and the group corresponding to the application you installed. For example, if you installed Jira Software, then the groups created are `jira-administrators` and `jira-software-users`. The `jira-software-users` group is also assigned as the default group for any new users created.

If you install and license further applications, Jira will create a default group for that application, but the Jira administrator will not be added to this group automatically.

Supplemental information:

See <https://confluence.atlassian.com/display/AdminJiraServer085/Managing+groups>.

# Giving Users Access to Applications



Jira Core



jira-core-users

1 Create User

2 Select Jira application

3 Added to default group



To log in and access a Jira application, a user must have application access. This is obtained by being a member of a group assigned to a Jira application.

When you create a new user, specify which Jira application(s) they'll be using. The user will automatically be added to the default group(s) associated with the application(s). In this example, Jira Core has been installed, and Jira has automatically created the jira-core-users group. You create the user Amit and select Jira Core as the application that he needs to access. Jira automatically adds Amit to the default group for Core, jira-core-users.

You can set the pre-selected default application(s) for new manually created or invited users (in the set defaults for new users on the application access page). Also, if you have more than one Jira application installed, it's possible to select which application or applications to which new users are automatically assigned. When you create a user, the applications you have selected as defaults will be pre-selected. However, it's possible to change this while creating the user. If you allow users to sign up via email, via public signup, or through a mail handler, they will be given access to the applications you select. We use groups to specify application access, rather than granting users access to applications individually, to make administration at scale easier. When you have thousands of users, you only want to work at the group level. Also, the default groups make it easy to give someone access to an application as these groups are already set up with the appropriate permissions.

# Application Access Example



You can specify which groups can access an application on the Application access Jira administration page.

You can have multiple groups for an application. In this example, both the jira-servicedesk-users group and the jira-administrators group can access Jira Service Management.

You can define more than one default group for an application. This means that if you create a user and specify an application with multiple default groups, the user will be added to all of those groups. In this example, Jira Software has two default groups – jira-software-users and Development – so any new Software users will be added to both groups, but will only consume one license.

Also, you can assign the same group to several applications. You may wish to do this to ensure members of this group always have full application access. In this example, the jira-administrators group has access to all three applications. Care should be taken when assigning a group to multiple applications, as the group members will consume a license for each application (unless they're already in a group assigned to that application). The exception is Jira Core. A user with access to any other application automatically has access to Jira Core, so they will not consume a license for Jira Core if they belong to a group associated with another application.

# Access to Multiple Applications

User is licensed for	Can see by default	Can see specific application features and functions
Jira Core	All projects	Jira Core
Jira Software	All projects	Jira Software and Jira Core
Jira Service Management	All projects	Jira Service Management and Jira Core



When you have multiple applications installed, by default, all users can view all projects (unless there are specific project permissions set up that prohibit this. We'll cover project permissions later). This means a Jira Core user will be able to see all Jira Software and Jira Service Management projects. However, as they are not licensed for these applications, they will not be able to see any features or functions that are specific to that application. For example, a Jira Core user viewing a Jira Software project would be able to see the project and its issues, but would not be able to see any Jira Software specific features, like Agile boards, development information, or release information. These features can only be viewed by a Jira Software user.

It's important to note that Jira Core does not have any specific features or functions that cannot be viewed or acted-on by other users. This means that if you are a Jira Software or Jira Service Management user, you can already view and work on a Jira Core project. You do not need to have specific application access for Jira Core, and therefore do not need to consume a license.

Supplemental information:

<https://confluence.atlassian.com/display/AdminJiraServer085/Jira+applications+and+project+types+overview>.

# Administrator Access to Jira Applications



## Access to all applications

- Their Admin account can do everything
- Easily test problems

## Administration only

- Cannot access application functionality (such as boards in Jira Software).



If a Jira administrator has application access, they can access that application's functionality. For example, if the administrator has Jira Software application access, they can access Kanban and Scrum boards and thoroughly test any problems that users are having. If they do not have application access, they can access issues in projects (depending on project permissions), but they cannot see application-specific functionality.

If the Jira administrator does need access to all application functionality, then the `jira-administrator`'s group needs to be added to all Jira applications' default groups. By default, the `jira-administrators` group will only be added to the default application group for the first application that is installed. Be aware that access to each application will use up a license seat.

Note that there is no way to lock a Jira administrator out from accessing issues or applications as they have the power to change project permission settings and also add their group to any application!

# How do we Bring Users into Jira?

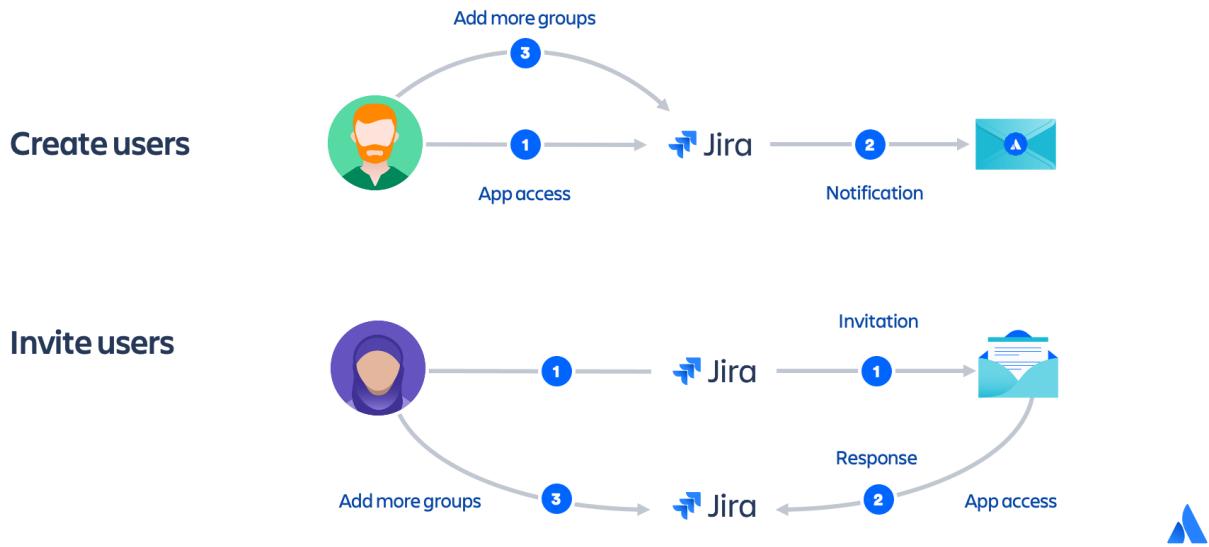
-  Manually create them
-  Send them an invite
-  Add an external user directory



There are three ways we can bring users into Jira:

1. We can manually add them
  2. We can send them an email invitation to create an account
  3. And we can add an external user directory such as LDAP or Crowd
- We'll cover each of these in the following slides.

# Creating vs. Inviting Users in Jira



There are two main ways to add users to Jira when public signup is not enabled.

To create users:

1. You can add the user and specify their application access – this determines their default group(s). If you're using the Jira Internal Repository or an external user repository such as LDAP set up for Read/Write, the Jira administrator would create the users. However, if you are using an external user repository set to read-only and you want all your users stored there, then the user would be created by the LDAP Manager. If you have the external directory set to Read Only, with Local Groups, you can create a user locally.
2. The user can optionally receive an email to let them know they've been added to Jira.
3. The Jira administrator can then add the user to any other required groups if necessary.

To invite users:

1. You can send an invitation and specify application access – this determines their default group(s)
  2. The user will be auto-added on response
  3. The Jira administrator can then add the user to any other required groups if necessary.
- Note that Jira's SMTP mail server must be configured to send notifications before you can invite users via email. Each invitation can only be used to create a user under the email address that it was sent to, and can only be used once. The invitation expires in 7 days. Your user license count won't be affected until users accept the invitation.

Creating a user manually in Jira is the only way you can set the user's password.

## User Creation vs. Invitation

	Advantage	Use Case
Creation	<ul style="list-style-type: none"><li>• Can add additional groups immediately</li></ul>	Enrolling a new member of the support team who needs to use Jira
Invitation	<ul style="list-style-type: none"><li>• Less information to add initially</li><li>• Only users who respond use up a license</li></ul>	Inviting the Sales team to register bugs reported by your clients

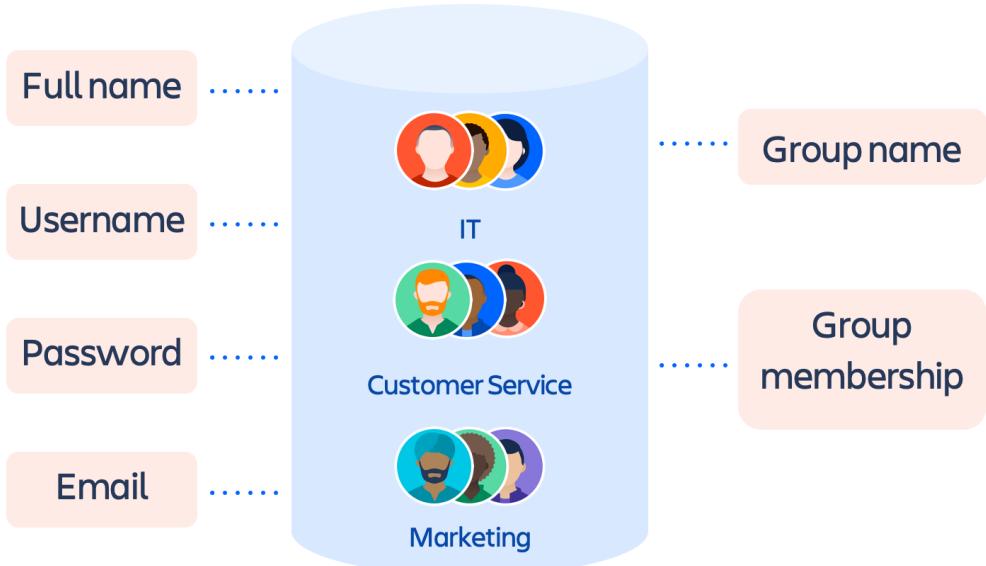


Whether to use creation vs. invitation is a balance. As the Administrator, you need to decide if it's right for your organization. Decide what sort of people need access. Some users will always need access, such as developers, and you would most likely add them via create. However, it might be the case that there's a situation where users need access as a sort of one-off. For example, the sales team might need access to a project, but only to that project and only as needed. You can invite them, and they might only use that account if they need it.

Some questions to ask that might help you decide are:

- Who needs access?
- Who may need access?
- What type of users need access?
- Are there some users who need minimal access?

## User Directories



A user directory provides persistent storage for information about users and groups. User information includes the person's full name, username, password, email address and other personal information. Group information includes the name of the group, the users that belong to the group, and possibly groups that belong to other groups.

## Example of Using External User Directories



“We have 10,000 users and need to manage our Jira users in our company’s existing LDAP.”



There are many combinations of where user data may come from. Jira comes with an inbuilt internal user directory, and this may be sufficient to store your user data. You can also add external user directories. In this example, this organization has 10,000 users. It uses LDAP for its Corporate directory, Crowd for single sign-on (SSO), and the Jira internal directory for temporary guest users.

The following types of directory servers and directory managers are supported:

- An LDAP directory for delegated authentication. A wide range of LDAP providers are supported, such as Microsoft Active Directory or OpenLDAP. An LDAP directory is a collection of data about users and groups. LDAP (Lightweight Directory Access Protocol) is an Internet protocol that web applications can use to look up information about those users and groups from the LDAP server. You can connect your Jira application to an LDAP directory for authentication, user and group management. Connecting to an LDAP directory server is useful if your users and groups are stored in a corporate directory.
- Atlassian Crowd - Groups are managed fully (unless they sit on top of LDAP). Crowd can collate multiple directories. Note, Crowd is not covered in this course. Crowd is the only supported SSO right now. You can use other SSO solutions, but Atlassian doesn’t support them.
- Another Jira server.

For more information, see the appendix in the Student Guide and Lab Workbook.

# Are you getting it?



Which statement is true?

- a. A user must have Jira Software application access in order to view issues in a Jira Software type project.
- b. The default group granted Jira Software application access must always be jira-software-users.
- c. More than one group can be granted Jira Software application access.



The answer is on the next slide.

## Did you get it?



Which statement is true?

- a. A user must have Jira Software application access in order to view issues in a Jira Software type project.
- b. The default group granted Jira Software application access must always be jira-software-users.
- c. More than one group can be granted Jira Software application access.



Answer: c.

a is incorrect because you can access issues in a project even if you don't have application access for that application.

b is incorrect because you can change the default group associated with Jira Software application access.

## See how it's done



- Users and groups
- Application access



In this demonstration we'll look at users and groups and we'll also look at application access and how that works.



## Takeaways

- Only have one default group assigned to each application
- Don't use the same group to grant access to multiple applications
- Admin groups should not be default groups
- Use the default group to allow application access, and create and manage other groups to control project specific permissions and access



Here are some of the headlines to take away from this module.

We strongly recommend that you only have one default group for each application. New users created and assigned application access are added to that application's default group. If this group is also assigned to an additional application, the new user will also gain access to the additional application.

For example, if you assign a group called GroupA to Jira Software and make it the default group, all new users added to Jira Software will also be added as members of Group A. If you then add GroupA to Jira Service Management, all users in Group A will now have full access to both Jira Software and Jira Service Management. This means that when you create a new user and add them to Jira Software, they will also gain access to Jira Service Management and consume a license for both applications.

If you have installed multiple applications, jira-administrators will only be added automatically to the first application installed/licensed. Being an administrator means you have access to the administration pages and can change settings and create projects. To access application-level functionality, such as Kanban or Scrum boards, you need to be in a group that has the appropriate application access.

We'll cover project-specific permissions later in the course.

# Try it

## Lab 3 - Setting Up Users & Groups



- Exercise 1 – Exploring Users and Groups
- Exercise 2 – Creating a New User
- Optional Exercise 3 – Deactivating a User & Exploring Access



In this lab, you'll cover setting up and managing users and groups.

4

# Configuring Global Permissions



In this module, we'll look at configuring global permissions.

# What will you learn?



- List the global permissions
- Determine which users have what global permission
- Assign global permissions to groups
- Remove global permissions from groups
- Observe the impact of different permissions



We'll look at how to determine which users have been granted specific global permissions.

We'll look at how you can grant and revoke global permissions to groups of users. We'll also look at each global permission and the implications of granting those permissions to groups of users.

# What are Permissions?

Control what users can do in Jira



Permissions are settings within Jira that control what users can see and do. Permissions are relevant to all Jira applications, and they exist at the global and project level. They control things like a user's ability to create new projects or whether a user can see or edit a comment on an issue.

Permissions are different from application access; application access grants groups Use access for an application.

# Global Permissions vs. Project Permissions



Control what groups of users can do system wide

A limited number of permissions

Granted to groups



Control what groups of users can do within a project

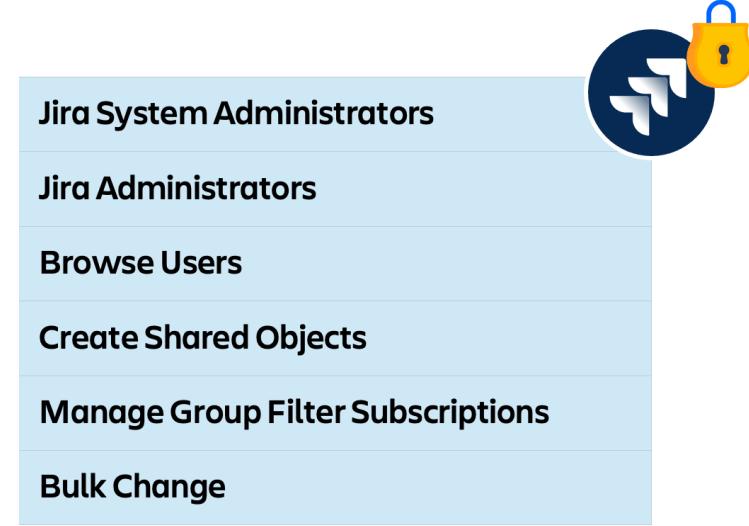
Many permissions

Granted to individuals, groups, and project roles



Global permissions allow you to control users' access to system-wide functionality. They can control, for example, whether users can share filters with other users, who can create projects, and who can perform bulk changes to issues. Users who have the Jira Administrators or the Jira System Administrators global permission can grant global permissions to groups of users. We'll look at a list of global permissions in the next slide. Project-specific permissions let you restrict project-related functionality to individual users, groups, or project roles. You can control, for example, who can see the project's issues, and who can create, edit, and assign them. Project-specific permissions and project roles are covered later in the course.

# Jira Global Permissions



Here are the 6 global permissions. We cover the 2 Administrator permissions next slide. Browse Users lets users @mention other users in text fields and comments in Jira issues. If users don't have this permission, they can view users or groups in most, but not all 'User Picker' menus and popup screens throughout Jira. E.g., sharing dashboards or filters. Create Shared Objects lets users share a filter or dashboard globally or with groups of users. It also controls who can create an Agile board.

Bulk Change: lets users modify a collection of issues at once. E.g., they could resolve multiple issues in one step. The bulk operations are Bulk Edit\*, Bulk Move\*, Bulk Workflow Transition, and Bulk Delete\* (\* this is all subject to project-specific permissions – we'll discuss those later.) Take care when granting the Bulk Change permission. E.g., a user with the Bulk Change global permission and the Add Comments project permission could comment on all of the issues that they can access. Undoing such modifications may require database changes (which we do not recommend).

Manage Group Filter Subscriptions lets users create and delete group filter subscriptions. A filter subscription is where you create a search filter, and then you set up a subscription for an individual or group. Those users receive an email containing a list of issues from the filter results as frequently as you require, even as often as every 15 minutes.

The Browse Archive global permission isn't listed in the slide because it's for Jira Data Center only. In Jira Data Center, it's possible to archive projects; users with this global permission can view issues in those archived projects.

# Administrator Global Permissions

## Jira System Administrators



- Has *all* administrative functions
- Cannot be assigned with Cloud

## Jira Administrators



- Has *most* administrative functions
- No functions affecting the application environment or network



People who have the Jira System Administrators permission can perform all of the administration functions in Jira. In contrast, people who have *only* the Jira Administrators permission cannot perform tasks that could affect the application environment or network. For example, users with the Jira System Administrators permission can configure Jira's SMTP mail server for notifications, and they can configure listeners. They can access logging and profiling information, and so on. This separation is useful for organizations which need to delegate some administrative privileges (such as creating users or creating projects) to particular people, without granting them full rights to administer the whole Jira instance.

For a list of the things that Jira system administrators can do that Jira administrators cannot do, see

<https://confluence.atlassian.com/display/AdminJiraServer085/Managing+global+permissions>. You look at this page in the lab.

If you only have Jira Administrators permission, you cannot grant yourself Jira System Administrators permission.

The administration group names and group permissions are very similar, so try not to get the Jira System Administrators and Jira Administrators global permissions confused with default groups in Jira.

# Separating Administrators



By default the jira-administrators group has both permissions



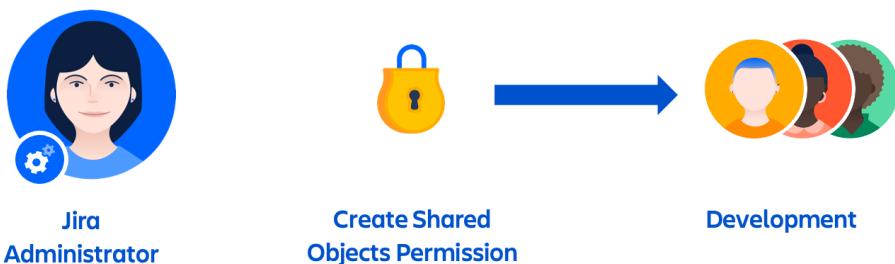
By default, the jira-administrators group has both Jira System Administrators and Jira Administrators global permissions. Also, by default, the user account created during Jira setup is a member of this jira-administrators group.

In your Jira instance, you might separate the groups who can access these permissions so that you reduce the number of “dangerous” or more profound changes to a limited number of users. Smaller companies may have one or two users in a group with both permissions, but at the enterprise level, they would probably separate them.

When you choose to separate administrators into different categories of permissions (Jira Administrators and Jira System Administrators), you are saying that some of your administrators should not be allowed to make specific system-wide configurations. This is useful if the expertise to make those changes resides with a small number of people. The downside to limiting the administrator privileges for Jira administrators is that they sometimes need assistance to configure Jira. Balance limiting privileges with how responsive system administrators can be to requests for support.

If you need some people to have only the Jira Administrators permission (and not the Jira System Administrators permission), you need to use two separate groups.

# Adding Global Permissions to Groups



Global permissions, as we have discussed, are assigned to groups. In this example, a user with the Jira Administrators permission assigns the Create Shared Objects permission to the Development group. Many people think that global permissions can be added to or further defined, but they can't.

Users granted the Jira Administrators or the Jira System Administrators permissions can assign global permissions.

## Global Permissions Assigned to Default Groups

Application	Group	Global Permissions
All	jira-administrators	Jira Administrators Jira System Administrators Browse Users Create Shared Objects Bulk Change Manage Group Filter Subscriptions
Jira Core	jira-core-users	Browse Users Create Shared Objects Bulk Change Manage Group Filter Subscriptions
Jira Software	jira-software-users	
Jira Service Management	jira-servicedesk-users	



These are the default global permissions that are assigned to the default Jira groups. If all of the users in Jira (apart from Administrators) are members of one of these default groups, they have four non-administration global permissions by default.

We'll discuss Administrator permissions more in the next few slides. When you install Jira, the default administrator, named admin, is part of the jira-administrators group, and so has both Jira System Administrators and Jira Administrators permissions.

# Are you getting it?



If a user has the Bulk Change permission (plus the project permission to delete issues) they could delete all the issues in a project.

- a. True
- b. False



The answer is on the next slide

## Did you get it?



If a user has the Bulk Change permission (plus the project permission to delete issues) they could delete all the issues in a project.

- a. True
- b. False



Answer: a True

## See how it's done



- View default global permissions
- Remove a global permission from a group



In the demo, we'll view the default global permissions, and we'll look at revoking a group's global permission.



## Takeaways

- Be careful granting Browse Users to a group that can access Jira externally
- Consider carefully who you grant Bulk Change to



Here are some of the headlines and best practices to take away from this module.

The Browse Users global permission is used for selecting users and groups in popup screens. This permission enables auto-completion of user names in most, but not all 'User Picker' menus and popups.

Consider the decision to grant the Bulk Change global permission carefully. This permission gives users the ability to modify a collection of issues at once. It's possible to set Jira to run in Public mode. When a Jira instance is in public mode, this means that anybody can sign up, and if they have the project permission to do so, they can create issues. A user with the Bulk Change global permission and the Add Comments project permission could comment on all of the issues that they can access. Undoing such modifications may not be possible through the Jira application interface and may require changes made directly against the database (which we do not recommend).

Permissions might get pretty complicated if many groups and roles are involved, so you might want to keep it as simple as possible.

Consider these implications when adding groups to global permissions. For example, if users can bulk update issue and they also have the project permission to delete issues, they can bulk delete issues.)

# Try it

## Lab 4 - Configuring Global Permissions



- Exercise 1 – Configuring Administrator Permissions
- Exercise 2 – Removing the Bulk Change Permission
- Optional Exercise 3 – Exploring Non-Administrator Permissions
- Optional Exercise 4 – Exploring the Jira Administrator Differences



In this lab, you will configure global permissions.

5

# Creating & Configuring Projects



In this module, we'll look at creating and configuring projects.

# What will you learn?

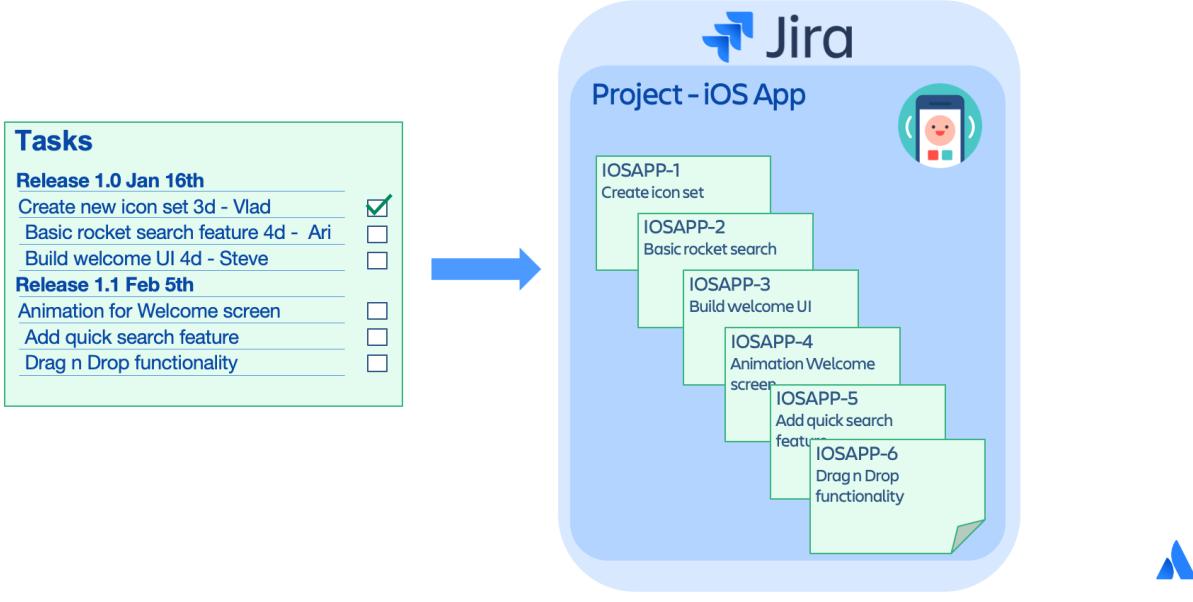


- Outline the advantages of project management in Jira
- List the different types of projects
- Create a new project
- Update the default project workflow
- Edit a screen to remove and move fields
- Add an issue type to a project
- Change the workflow for an issue type



In this module, we will cover creating and managing projects in Jira.  
We'll cover....

# Project Management with Jira



Jira helps out with the challenge of project, team, and task management. In Jira, projects are created and associated with tasks, resources, and schedules. An issue represents a task in Jira. Projects are a collection of issues, which are the building blocks of any project. Issues act as packets of work that travel through their respective workflows within their projects until the work is completed.

All of these tasks need to be managed and monitored. Jira has in-built features and functionality that help you to manage these projects, the people involved in them, and the tasks that need to be completed.

In this example, we are working on a software development project to create an iPhone app for Teams in Space. It allows all life forms to book travel from Mars. Teams are located across the universe, and Jira is the ideal solution to help the team collaborate. 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. A Jira project can also be configured and customized to suit the needs of you and your team.

# Review of a Jira Project

Terminology		
		<b>Team members</b>
		<b>Tasks</b> Bug, task, invoice, etc.
		<b>Processes</b> Processes for each task
		<b>Releases</b> 1.0, 1.1, 1.2, etc.
		<b>Software components</b> UI, database
		<b>Data</b> For reporting on project status & metrics



Team members or users manage and work on the tasks in the project. These tasks are managed in issues. In a project, there are different types of issues for managing things like bugs, stories. These are different because they have different business processes behind them and gather different kinds of information necessary to complete them. Jira projects are a way of grouping issues together and a way of applying the same sets of configurations to issues. These configurations, such as workflow, issue types, and screens, can be changed on a per-project basis so that each project can have a different set of configurations, or the configurations can be shared.

Data needs to be captured, such as which version of iOS a bug manifests itself or what status the task is in (such as open, or complete). Data is automatically gathered to provide reports on project status and metrics.

Workflows are your business processes. A workflow defines the sequence of steps that all issues of a particular issue type must go through to completion. Issues have associated workflows because different tasks have different processes associated with them.

Versions define a release version, milestone, or a phase pending on the business process. Issues can be packaged together and released in a version. Versions let you categorize content to manage the tasks. Software projects typically use versions.

Components define sub-categories or groupings of the project. Issues can belong to software components such as UI or database.

# Project Types



◆ Jira Software



⚡ Jira Service Management



↑ Jira Core



Each of the three Jira applications is suited to different types of projects.

Jira Software is ideal for creating software development projects. For example, to develop a new Cloud application.

Jira Service Management is ideal for any Help Desk application. For example, an IT Help Desk.

Jira Core is ideal for any business application, for example:

- HR uses it to manage candidate tracking and employee onboarding.
- Finance uses it to manage tracking and procurement processes.
- Marketing uses it to manage review cycles and marketing collateral.

# Project Templates

The screenshot shows the Jira Project Templates interface. It is organized into three main sections: Software, Service Desk, and Business. Each section contains two or more project templates with icons and descriptions.

Category	Template	Description
Software	Scrum software development	Agile development with a board, sprints and stories. Connects with source and build tools.
	Basic software development	Track development tasks and bugs. Connects with source and build tools.
Service Desk	Basic	Track, prioritize and resolve your organization's requests, everything from IT to HR to finance.
	Customer service	Provide support, collect feedback, and track your external customers' satisfaction.
Business	Project management	Plan, track and report on all of your work within a project.
	Process management	Track all the work activity as it transitions through a streamlined process.
	IT Service Desk	Manage incidents, changes, problems and service requests with ITSM workflows.
Task management	Quickly organize and assign simple tasks for you and your team.	

Depending on which Jira applications you have installed, you may have more than one project type available. The project types are Software (if you have Jira Software installed), Service Desk (if you have Jira Service Management installed), and Business (if you have Jira Software, Service Management, or just Jira Core installed). Each project type has specific features.

Each project type has several project templates available, as you see here. Project templates are sets of pre-configurations, which are the default starting points.

For example, Software projects give you the following templates:

- Scrum software development gives you a template for Scrum software development, including a Scrum board and Agile-specific issue types, like Story and Epic.
- Kanban software development gives you a template for Kanban software development, including a Kanban board and Agile-specific issue types, like Story and Epic.
- Basic software development gives you a template to use for software development. Once you create a project, users can go to work with the defaults, or you can customize them to suit your needs.

You can convert a project from one type to another (as long as the application is installed.) For instance, you can convert a Jira Software project to a Jira Core project at the end of a Jira Software evaluation period, or when your team grows. You can only convert to project types of Jira applications that you have installed.

# What Administrators Need to Create & Access Projects

What you need	to do...
 Jira Administrators global permission	Create projects
 Application Access Project permissions	View the project you create



As the Jira administrator, you act as the project creator for your Jira community. A user who has the Jira Administrators global permission, the Jira administrator, can create projects for all applications installed. If the administrator does not have application access for that application, they can not use project-related functionality in the project after they have created it. If you're creating a project using a project type related to an application you currently do not have access to, Jira displays a checkbox that allows you to grant yourself access to that application. This action adds you to the default group of that application, and you count as a user for that license.

You also need the appropriate project permissions for the project, for example, the Browse Projects permission. We'll discuss project permissions in the next module.

Supplemental information:

<https://confluence.atlassian.com/display/AdminJiraServer085/Defining+a+project>.

# Creating a New Project

The screenshot shows the 'Details' section of the Jira 'Create Project' dialog. It includes fields for Name (Human Resources), Key (HR), URL, Project type (Business), Project category (None), Avatar (a small circular icon), and Description (Human Resources tasks to manage Teams In Space employees across the universe). Blue dotted lines connect specific fields to corresponding tips on the right:

- Key field is connected to the tip: "Choose a short, descriptive key".
- URL field is connected to the tip: "If available, provide URL to project docs".
- Project type dropdown is connected to the tip: "Categorize projects".
- Avatar icon is connected to the tip: "Use a meaningful icon".
- Description field is connected to the tip: "Provide a description to help identify projects".
- A blue dotted line also connects the top-right corner of the dialog to the tip: "Use a descriptive name".

When creating a project, use a descriptive name for the project, so it's easy to find. The project key will become the prefix of this project's issue keys (e.g. 'HR-100'). Choose one that is descriptive and easy to type. The key can be changed, but it is not a trivial task. By default, the project key is the first letter of each word of the project's name. If available, provide URL to Confluence space, document repository, etc. for project documentation.

If you have a large number of projects, you can use project categories to categorize them. We'll discuss this later in this module.

An Avatar (icon) makes the project easier to find. You can use one of the provided avatars or an image of your choosing. If you don't want to use a project avatar, you can upload a transparent pixel, which makes the avatar invisible.

As a Jira administrator, you need to manage many projects, including many in which you are not involved. Adding a description makes it clear what the purpose of the project is. In the URL field, you could add a link to any supporting documentation, for example, a Confluence space, where users can find more information relating to the project. You can include HTML in the project description, but make sure all your HTML is valid. Please be aware that this is completely unfiltered HTML, and as such, it is susceptible to cross-site scripting attacks.

# Issue Types for Projects

## Scrum & Kanban software development

-  Bug
-  Task
-  Sub-task
-  Story
-  Epic

## Basic software development

-  Bug
-  Task
-  Sub-task
-  Improvement
-  New Feature
-  Epic

## Business (Core)

-  Task
-  Sub-task

## Jira Service Management

-  IT Help
-  Service Request
-  Service Request with Approvals
-  Task
-  Sub-task

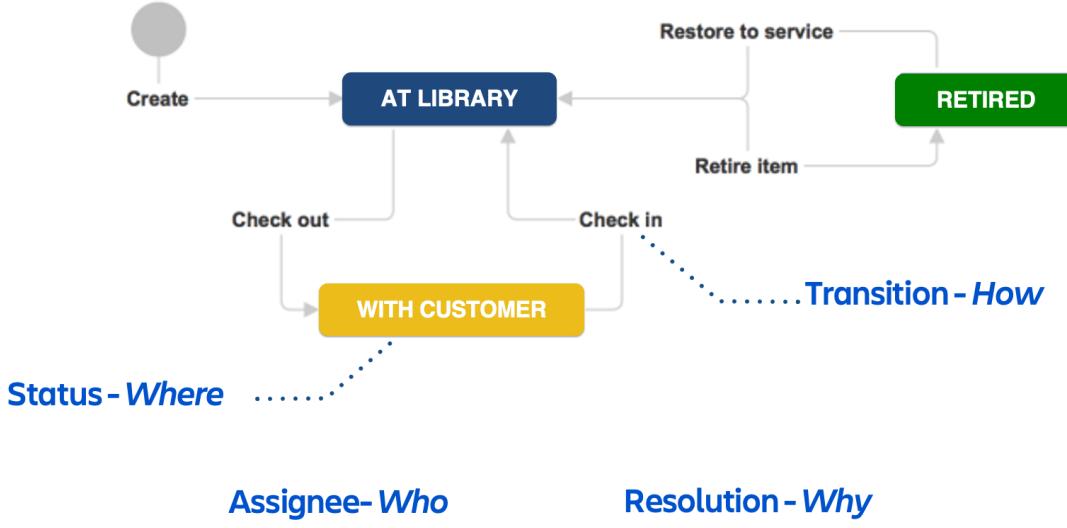


Jira enables you to keep track of different types of tasks such as bugs, tasks, helpdesk tickets, and so on by using different issue types. When creating a project, project-specific issue types are created by default, and the Jira administrator can add/edit Issue types as necessary to support the business process. You can also configure each issue type to act differently (e.g. to follow a different process flow or track different pieces of information). In this slide, you see the default issue types for Software, Core, and basic Service Management projects. On installation of Jira, you only see a few issue types; more are added as you create projects.

Here is a list of the default issue types (for Core it's just Task and Sub-task):

- Task – Work that needs to be done.
- Sub-task – A piece of work that is required for a task.
- Bug – A problem that impairs or prevents the functions of a product.
- Story – The work to develop some deliverable features.
- Epic – A big user story that needs to be broken down.
- Improvement – An enhancement to an existing feature.
- New Feature – A new feature of the product.
- Service Management Issue Types (there are more for other types of service desks):
  - Service Request – General request from a user for a product or service.
  - Service Request with Approvals – Service request requiring approval.
  - IT Help – Requesting help for IT-related problems.

# What Makes Up a Workflow?

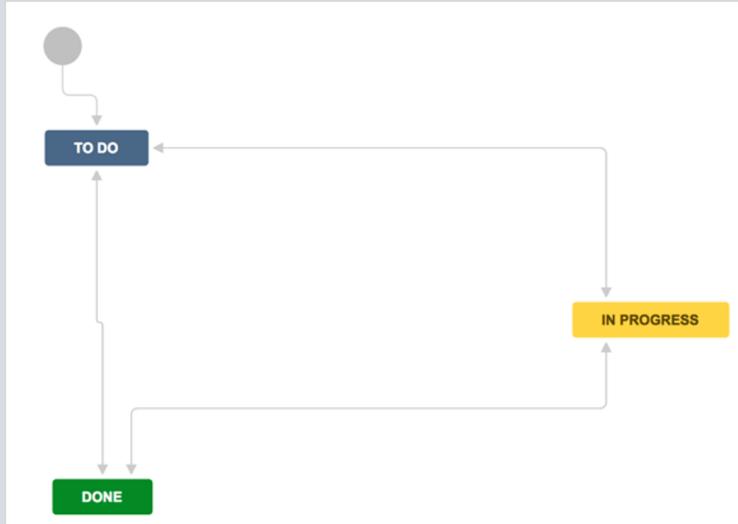


Here's an example workflow. An item managed in a library, e.g. a book or a magazine, could be managed in Jira as an issue and administered in the workflow shown here. A workflow has four components: statuses, transitions, assignees, and resolutions.

- Statuses represent the position of the issue in a workflow. A book can be at the library, with a customer, or retired from inventory. Every status should be a unique state in your workflow and describe what's already happened. Keep statuses to a minimum.
  - Transitions are the way a particular issue moves from status to status. Books are loaned out by librarians, returned by customers, and evaluated by staff to see if they're fit for circulation. Jira sets workflow permissions that allow only the right people to transition an issue. You can also use screens to gather data during transitions.
  - Assignees - Workflows guide how people work together. In Jira, the assignee dictates who's responsible for an issue. When books are loaned out, the assignee is set to the borrower. When the book is returned, the assignee is removed when it's checked in.
  - Resolutions explain why an issue transitions from an open status to a closed one. Retiring a book removes it from inventory; therefore, we can put a resolution on that book, such as damaged, out of date, lost, and so on.
- A common mistake is to confuse resolution with status. Status describes where an item is in the workflow; resolution explains why an item is not in flight anymore. Another common mistake is setting resolution incorrectly. You must set the resolution when an issue moves from an open state to a closed state.

# Default Workflows for Projects

Core Project Management



Each template for each project type (Software, Core, and Service Management) has a different default workflow. In this example, you see the default workflow for the Core Project Management template. It's a simple workflow with issues typically flowing from TO DO to IN PROGRESS to DONE. But note the double-headed arrows meaning issues can also flow from DONE back to TO DO or IN PROGRESS, and from IN PROGRESS back to TO DO.

## Simplified Workflow



This is the default workflow for Software projects using the Scrum template. This is a simplified workflow, which means that any status can transition into any other status.

# Editing & Creating Workflows

- You can't modify a 'live' workflow – instead edit a draft then publish
- Always save a backup copy when publishing workflows
- Create a new workflow by:
  - Cloning an existing workflow
  - Creating a new workflow
  - Importing a workflow
- Test on a staging environment



When you create a project, the project gets a copy of the workflow for the project type and template. So you can change it, and the changes only apply to that project. However, you can't modify a 'live' workflow; you can only edit a draft copy. Then you publish the draft, which replaces the live workflow. You also get the option to save the original workflow as a backup copy. It's always a good idea to do this in case you ever want to revert to the original workflow or use it somewhere else.

You can create a new workflow using one of these methods:

- Cloning an existing workflow – copy the live workflow, then edit the copy.
- Creating a new workflow.
- Importing a workflow – either locally or from the Atlassian Marketplace.

We won't cover advanced workflow options in this class, like conditions, validators, post functions, and transition screens. See the Getting More from Jira Workflows course for more on advanced workflow options.

If you test in a staging environment, you won't make an impactful change to configurations or data that could impact your production environment. Test there first, then test in production.

The number of issues in use in your Jira instance impacts the speed of configuring a workflow. If you have thousands of existing issues, this process may take some time. And once this process begins, it cannot be paused or canceled. Please avoid editing or transitioning any issues within your project while this process is taking place.

# Are you getting it?



What do you need to create projects?

- a. Application access for the type of project you want to create
- b. Jira Administrators global permission
- c. All of the above



Answers on next slide

## Did you get it?



What do you need to create projects?

- a. Application access for the type of project you want to create
- b. Jira Administrators global permission
- c. All of the above



Answer: b, you only need Jira Administrators global permission to create projects.



## Takeaways

- Before creating a workflow, gather all your stakeholders
- Keep the workflow simple – less is more
- Don't confuse resolution with status
- Always save a backup copy when publishing edited workflows



When you build a process around a set of people, identify all the stakeholders and consult them before creating the workflow. E.g., if you're building a workflow between product management, software development, and support, ensure you have one representative from each team in the meeting. Talk to them about what's important to them. Before the meeting, draw a draft workflow on the whiteboard, then walk through each use case in the meeting and gather stakeholder feedback. Use a whiteboard, so it's easy to get feedback and make changes at this stage.

Keep the workflow simple. Often stakeholders want to have statuses for every part of their business process. Each status adds more transitions and complexity to the workflow. Whenever adding a new status to a workflow, ensure that it is needed and adds value rather than adding unnecessary complexity into the flow. For example: The test manager wants a new status, Rejected, for all issues that don't pass review. This status isn't necessary unless you need it for reporting. Instead of adding a new status, you could add a transition named Rejected from the testing status back to the initial status. Alternatively move the issue into a Closed status and set the resolution to Rejected. You could also add a transition to re-open the issue from the Closed status. Users often confuse resolution with status. Status describes where an item is in the workflow; resolution explains why an item is not in flight anymore.

When you publish workflow changes, save the original workflow as a backup copy. With a backup in place, you can revert to the original workflow or use it somewhere else.

# Try it



## Lab 5 - Creating & Configuring Projects

- Exercise 1 – Creating a Project
- Exercise 2 – Updating the Workflow for a Project



In this lab, you will work through exercises that guide you through the process of creating and configuring projects.

# Screens & Fields

The screenshot shows a Jira issue page for a story titled "Extend booking experience in UI to include multiple hotels on one reservation". The page includes sections for Details, Description, People, and Dates.

**Details:**

- Type: Story
- Priority: Critical
- Affects Version/s: 1.5
- Component/s: Accommodations, Web Site
- Labels: None
- Epic Link: Large Team Support
- Story Points: 9

**Description:**

Currently LodgingController makes an assumption that all the participants in the group are on the same itinerary. Many of our hotel travel providers limit reservations to 8 people. To accommodate larger groups we need to have multiple hotel/motel reservations on one group reservation. Customers need to be able to select a number of hotels (or several reservations at the same hotel). Customers should not be able to book the reservation until all participants have a hotel room.

**People:**

- Assignee: Kevin Campbell (Assign to me)
- Reporter: Jennifer Evans
- Votes: 0 (Vote for this issue)
- Watchers: 0 (Start watching this issue)

**Dates:**

- Created: 25/Nov/18 11:01 AM
- Updated: 27/Nov/18 2:08 PM

A screen is mapped to a specific issue operation

Fields hold the information for an issue

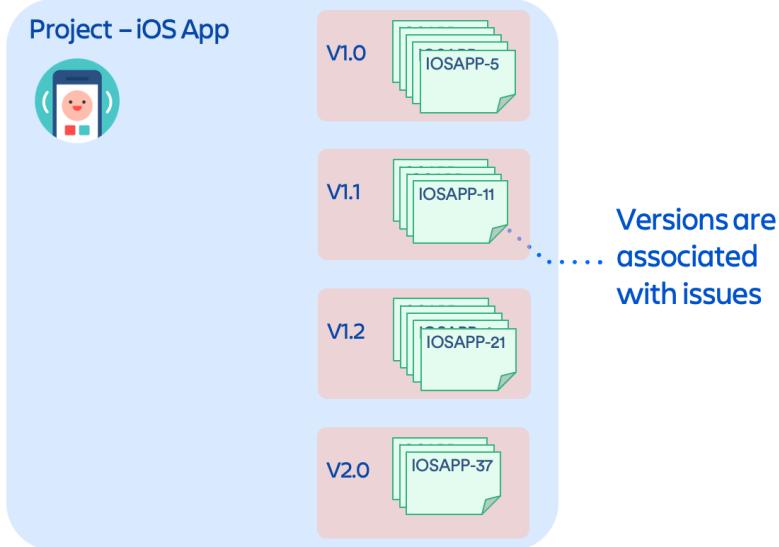
Users view, add and edit data in fields

A screen is the user's view of an issue and is simply a collection of fields. An administrator can define which screen displays when an issue is being created, viewed, edited, or transitioned through a particular step in a workflow. Here you can see an issue with a screen displaying in the view operation.

Information for each issue is held in the fields that are associated with that issue. You can tailor these fields to suit your organization's needs.

# Versions

- Allow repeated iterations of a project
- Are used to track release progress and risks



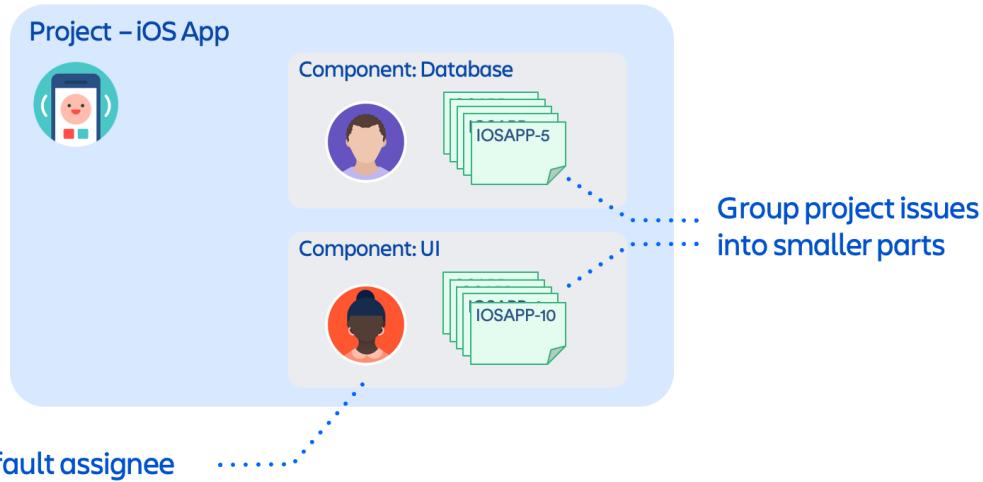
In Jira, you can group issues in a project using versions. For example, if you are using a Jira project to manage the development of a product or manage the build of a house, you may want to define different versions to help you track which issues relate to different phases or iteration of your product or build (for example, 1.0, 1.1, 1.2, 2.0, 2.0.1).

Jira can help you manage, release, and archive your versions. Versions can also have a release date and are automatically highlighted as "overdue" if the version is unreleased when this date passes. Versions are defined in the project settings. They help build out the roadmap for the project.

Versions are typically used in Software projects.

Project administrators manage versions; we'll discuss project administrators in the next module.

# Components



Components provide a way to group issues logically within a project. If you don't want to use too many projects in Jira, components are a great way to group related issues within a project.

In this example, we're developing an iOS app and have a Software project. An App might consist of components such as UI, Database, User Settings, and so on. Note that issues can belong to more than one component.

When configuring the component, you can optionally define a component lead; this is useful if you have different people leading different teams in your project. Typically you have a lead developer to manage the tasks related to those components. In Jira, you can easily add components and set up component leads so that the team member who manages issues of this type gets assigned the correct issues.

If an issue has multiple components on creation, the issue assignee will automatically be defined as the lead of the component that is listed first alphabetically. For example, Jane creates an issue and adds the components of UI, Database, and User Settings. She does not assign the issue to anyone. In this scenario, each component has a component lead, and each component lead is a default assignee. The default assignee for the issue will be the component lead for the Database component.

Supplemental information:

<https://confluence.atlassian.com/display/AdminJIRAServer085/Managing+components>.

# Project categories

View work across related projects in searches, filters, reports, etc.



Jira does not support sub-projects or parent projects



As a Jira administrator, you can create project categories so your team can view work across related projects in one place. Your team can use categories in advanced searches, filters, reports, and more. Note that project categories cannot be used to create project hierarchies (such as parent projects). A Jira project can only belong to one category. Jira does not support sub-projects or parent projects; that is, you cannot nest projects. In this example, there are two project categories - HR and App Dev. The HR category is associated with two projects - Guidelines and People. The App Dev category is associated with three projects - Mars App, Star App, and Sun App.

# What are schemes?

A scheme is a collection of configurations



Schemes save you work by re-using configurations across projects



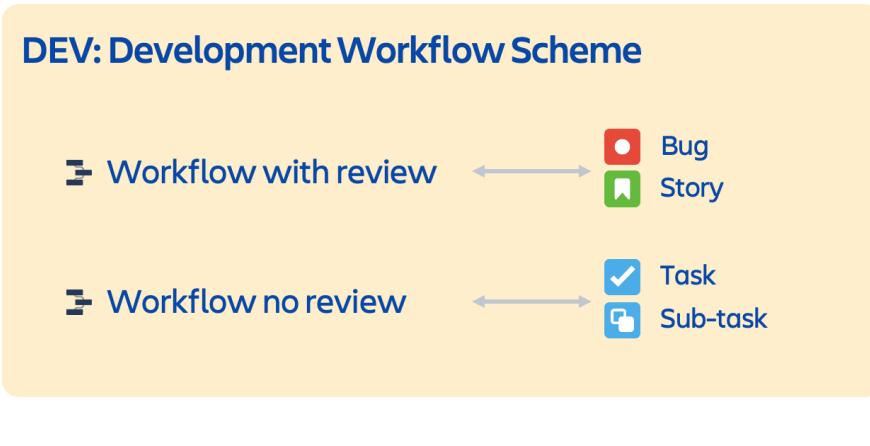
Schemes are probably one of the biggest sources of confusion for somebody relatively new to Jira administration. However, once you understand the core concepts, they are a great way to reduce administrative overhead while keeping flexibility up.

A scheme is a container or collection of configuration items. There are many different types of schemes, and each one behaves slightly differently. Schemes allow you to package up configurations and then re-use them in other projects.

In the example in the slide, you see an issue type scheme for Software type projects, which is named Dev Issue Type Scheme. It contains all the default issue types for that type of project. You can see the Jira administrator has added the Feature Request issue type. Then the Jira administrator associated the scheme with all their development projects. If you didn't use schemes, you'd have to configure each project individually. That could be a lot of work, and you'd have to repeat it every time you created a new development project.

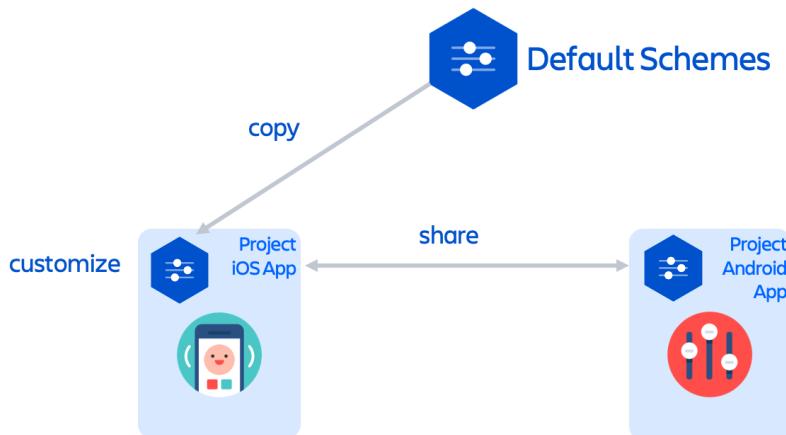
# Workflow schemes

- Schemes define which workflows apply to which issue types in a project



Here's another example of a scheme. A workflow scheme allows you to define which workflow to use with an issue type in specified projects. Here the DEV: Development Workflow Scheme is used in the DEV project. There are two workflows in this scheme - Workflow with review, which is applied to Bug and Story issue types and Workflow no review, which is applied to Task and Sub-task issue types.

# Sharing Schemes



Each new project gets its own fresh set of schemes based on the default schemes for its project type and project template. For example, if you created the iOS App project using the Scrum Software Development template, the project gets its own workflow scheme, which is a copy of the default workflow for this type of project. You can then customize these schemes without changing the default schemes. The schemes copied are for issue types, workflows, screens, and issue type screens. The schemes that are not copied are the permission schemes and notification schemes. If you want to have these schemes customized for your project, you need to copy them first manually. One exception to this is the field configuration scheme, which is automatically copied when a Service Desk project is created. Schemes are useful because you can share them between projects. Once you've configured a project the way you want it with your own issue types, workflows, and so on, you may wish to use that configuration in another project. Schemes let you do that. In the example shown here, when you create the iOS App project, it's created with a copy of the default schemes. You can then customize those schemes, for example, adding some issue types to the issue type scheme and a new workflow to the workflow scheme and so on. Then you want to might want to create a similar project for the Android App. When you create that project, you can choose to share the iOS App project's schemes, thus saving you all the work of customizing the issue types and workflow again.

This is just a brief introduction to schemes. We'll cover schemes in more detail later.

# Are you getting it?



A screen is mapped to a:

- a. Project component
- b. Version
- c. Specific issue operation



The answer is on the next slide.

## Did you get it?



A screen is mapped to a:

- a. Project component
- b. Version
- c. Specific issue operation



Answer: c, a screen is mapped to a specific issue operation such as creating an issue or editing an issue.

## See how it's done



- Create and view a Software project



In this demonstration, we'll create a Jira Software project and then take a look around the project's settings.



## Takeaways

- Don't create too many fields on a screen
- Think carefully before deleting fields
- Create project categories to help find projects
- When creating a new project that is similar to another, share its schemes



Here are some best practices to take away from this module.

Don't have a lot of fields display on a screen. Too many fields mean users probably leave half of them empty or possibly filled with random text, and you can't drive business on bad data.

Another best practice relating to fields is to be careful when removing fields. If you remove a field from a screen, you will not delete the existing data, nor will it affect search or reporting. However, if you delete a field, it destroys the data in the field.

Project components are a great way to group and categorize issues. Depending on your requirements, rather than create multiple projects, you can use components to differentiate and group issues.

Project categories are different from components. Why are project categories useful? Jira can search for all the issues across projects that have the same category. For example, if several projects have the "App Dev" category, you could use it in an advanced search and display issues sorted by the project category.

We'll cover more on sharing project configuration schemes in the next two modules.

# Try it



## Lab 5 - Creating & Configuring Projects

- Exercise 3 – Updating Fields & Screens in a Project
- Exercise 4 – Adding an Issue Type to a Project & Changing its Workflow
- Optional Exercise 5 – Updating a Workflow & Viewing Activity



In this lab you'll create and configure projects.

# 6

# Configuring Project Roles & Permissions



In this final lab, we'll look at configuring project roles and permissions.

# What will you learn?



- List the advantage of using roles
- Assign the project administrator role to delegate some administration responsibilities for a project
- Assign the project lead and default assignee roles for a project
- Create a project role and add members so you can assign permissions to users in a project
- Edit a new permission scheme using roles to control what users can do in a project



We look at....roles and the advantages of using them.

We'll cover assigning the project administrator role to delegate some administration responsibilities for a project.

We'll assign the project lead and default assignee roles for a project.

Then we'll create a project role and add members so you can assign permissions to users in a project.

And finally, we'll look at editing a new permission scheme using roles to control what users can do in a project.



# Permissions for Projects

- Apply only to projects
- Project-focused functionality
  - Who can create issues, comment on issues, etc.
- Granted to users, groups, or project roles



The features users can see, and the actions they can take in a project are determined by their application access and the project-specific permissions. Jira allows you to control who can access your project and exactly what they can do in that project, by using permissions for projects. Recall that global permissions apply to the entire Jira instance and are granted to groups. Global permissions are for independent project functionality such as who can administer Jira, perform bulk updates, and so on.

Permissions for projects are much more granular than global permissions and break down into different types:

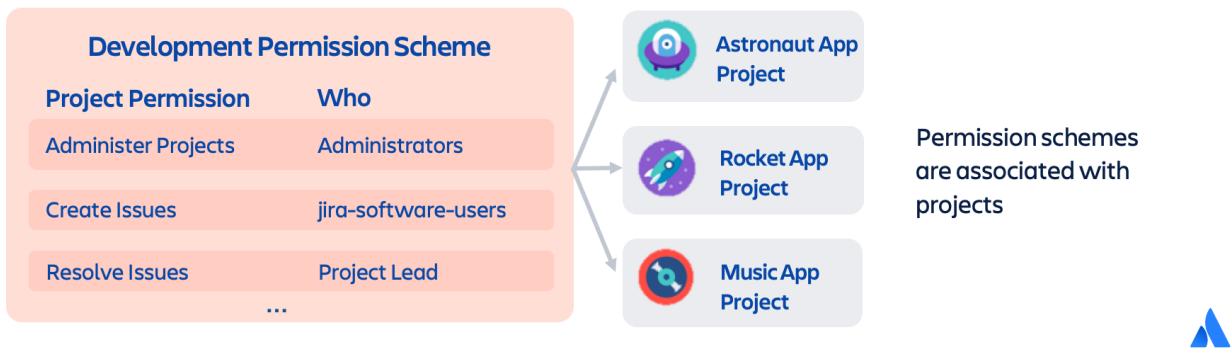
- Project Permissions such as Administer and Browse projects
- Issue Permissions such as Create and Resolve Issues
- Voters & Watchers Permissions
- Comments Permissions
- Attachment Permissions
- Time Tracking Permissions

For a list of all the permissions, see <https://confluence.atlassian.com/display/AdminJIRAServer085/Managing+project+permissions>.

You can also control access to individual issues by using security levels and issue security schemes. These are beyond the scope of this course. See <https://confluence.atlassian.com/display/AdminJiraServer085/Configuring+issue-level+security>.

# Permission schemes

- Permissions are organized into permission schemes
- Map project functionality (WHAT) to people who can perform them (WHO)



Project permissions are defined in permission schemes that are then associated with projects by the Jira administrator. The project's permission scheme determines who can do what in a project. Here is an example of a Permission Scheme, the Development Permission Scheme. As you can see, it associates groups (jira-software-users) and project roles (Administrators and Project Lead) with project permissions.

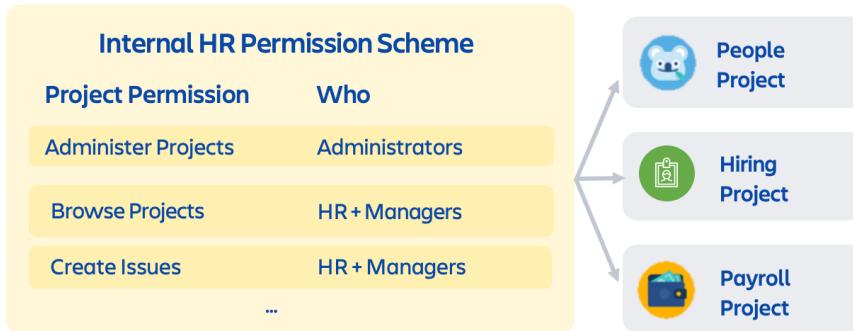
Permission schemes allow you to create a set of permissions and apply them to any project. All permissions within a scheme apply to all projects that are associated with that scheme. This approach enables flexible, fine-grained permission control.

In this example, the permission requirements in all development projects are the same. Members of the Administrators role can administer all DEV projects, only members of the jira-software-users group can create issues, and only the project lead can resolve issues. Teams in Space reuses this scheme in all of their dev projects.

Project permissions can be granted to individual users, groups, project roles, issue roles such as 'Reporter', 'Project Lead', and 'Current Assignee', and 'Public' (to allow anonymous access). Typically we don't recommend associating permissions with individual users.

# Advantages of using permission schemes

- Consistent permissions across projects
- No need to re-create permissions for every new project



In many organizations, multiple projects have the same needs regarding access. For example, only the project team can assign and work on issues. Permission schemes prevent the need to set up permissions individually for every project. Once a permission scheme is set up, it can be applied to all projects having the same requirements. As a Jira administrator, you'll create schemes that match your company-wide processes, or project types. You'll have a number of projects, and each will belong to a certain category of project, such as a product dev project, or a client project, or an internal project such as finance, marketing, or sales. Each time you set up a new project, you can choose the scheme that reflects the type of project you set up. This is useful to you and your users as:

1. You can manage the number of schemes and keep them to a minimum. Too many schemes will make your life as a Jira administrator painful.
2. Users will know how to work within the project as it's the same as each of the others, so you don't need to educate them with each new project that you set up.

In this example, we have an Internal HR Permission Scheme. They have restricted certain permission like Browse projects and Create issues to just members of the HR group and Managers. This scheme has been applied to a number of HR projects that have the same requirements.

# Default Permission Schemes by Application



Name	Projects
<b>Default Permission Scheme</b> This is the default Permission Scheme. Any new projects that are created will be assigned this scheme.	<ul style="list-style-type: none"><li>• Human Resources</li><li>• Legal</li></ul>
<b>Default software scheme</b> Default scheme for Software projects.	<ul style="list-style-type: none"><li>• Android App</li><li>• iOS App</li></ul>
<b>JIRA Service Desk Permission Scheme for Project BSD</b> This JIRA Service Desk Permission Scheme was generated for Project BSD	• Basic Service Desk
<b>JIRA Service Desk Permission Scheme for Project CSD</b> This JIRA Service Desk Permission Scheme was generated for Project CSD	<ul style="list-style-type: none"><li>• Customer Service Desk</li></ul>
<b>JIRA Service Desk Permission Scheme for Project ISD</b> This JIRA Service Desk Permission Scheme was generated for Project ISD	<ul style="list-style-type: none"><li>• IT Service Desk</li></ul>

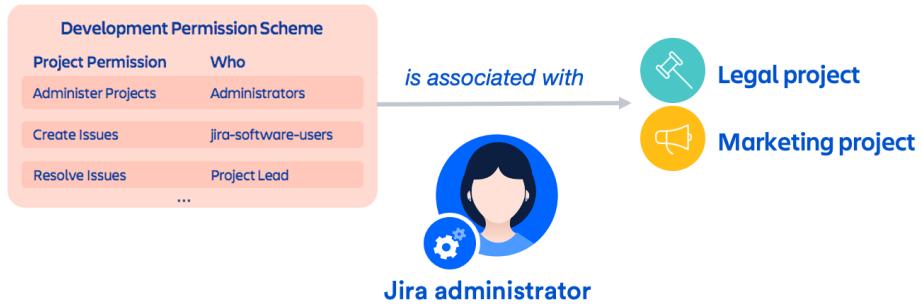
When you install Jira, it automatically creates the Default Permission Scheme. This permission scheme is used for and shared by any new Core (Business type) projects. If you have Jira Software installed and create a Software project, Jira creates the Default software scheme, which will be used for and shared by any new Software projects. Jira Service Management is different and creates a separate scheme for each Service Management project created.

Here you see both Human Resources and Legal projects share the same Default Permission Scheme, and both Android App and IOS Application projects share the default software scheme. The HR Help Desk and IT Help Desk projects have their schemes.

# Who can work with schemes?

Only Jira administrators can:

- Create and modify schemes
- Associate schemes with projects



Only Jira administrators can modify schemes and associate them with projects.

In this example, the Jira administrator has updated the Development Permission Scheme and is associating it with the Legal and Marketing projects.

# Project roles

- All projects have the same roles available to them
- Project administrators assign members to project roles for their projects

The screenshot shows the 'Users and roles' page in Jira. At the top, there are buttons for 'Add users to a role' and 'Edit defaults'. Below this, under 'Defaults', it shows 'Project Lead' assigned to 'Alana Grant' and 'Default Assignee' set to 'Project Lead'. The main section is titled 'Users by role'. It has a search bar with 'Roles: All' and a 'Search by name or email' input field. Under 'Administrators', it says 'Showing 1 of 1' and lists one user: 'jira-administrators'. Under 'Members', it says 'Showing 1 of 1' and lists one user: 'Sophie Nguyen'.

Name	License	Username	Email address	Last session
jira-administrators				

Name	License	Username	Email address	Last session
Sophie Ngu...	Jira Core	snguyen	snguyen@team...	3 days ago 4:20 AM



Next, we'll look at project roles. Roles are used in permissions, notifications, and issue security. Roles are also used with workflows and in other areas of Jira functionality, such as when sharing dashboards and filters.

In the slide, you can see the Users and roles page for a project. There are two roles; Members and Administrators. The jira-administrators group is associated with the Administrators project role for this project. Sophie Nguyen is a member of the Members role for the same project.

Project roles are a flexible way to associate users and groups with projects. In this project, Sophie has the Members role; in another project, she may have the same or different roles. Jira administrators create and define project roles. Once a Jira administrator creates a project role, all projects will have those project roles available to them.

Project administrators assign members to project roles at the project administration level.

# Project Roles vs. Groups



- Membership is for one project
- Only Jira administrator can create
- Project administrators can alter membership



- Membership is global
- Only Jira administrators can create
- Only Jira administrators can alter membership



Let's compare groups and project roles in Jira.

Project roles are similar to groups; the main difference is that group membership is global, whereas project role membership is project-specific. Groups or individuals are associated with roles on a project-by-project basis.

Group membership can only be managed by Jira administrators, whereas Project administrators can alter project role membership.

If you use groups in schemes, they cannot easily be shared between projects with unique project members. When using groups, a Jira administrator has to manage any permission scheme-related requirements, which increases the maintenance overhead.

Project roles, on the other hand, are a flexible way to associate users and/or groups with particular projects. A project role is a kind of a bucket that holds individual users or groups.

Project roles can be used in many places, including permission and notification schemes, issue security levels, workflow conditions, and comment visibility. Project roles can also be given access to issue filters and dashboards. The core usage for project roles is for permission and notification schemes. While you could assign permissions and notifications to users and to groups directly, roles are more flexible and sustainable.

Supplemental information:

<https://confluence.atlassian.com/display/AdminJiraServer085/Managing+project+roles>.

# Default project responsibilities



## Project Lead

- Manages the project
- Used in schemes and more

## Default Assignee

- Project Lead, Component Lead or Unassigned
- Issue Assignee is set to this if no one is assigned when the issue is created

## Project Administrator

- Configure project
- Assigns project members to project roles
- Define components and versions
- View the project's schemes
- jira-administrators by default



Different people may have different responsibilities in various projects. You can assign users to some responsibilities in your project. Each project has a Project Lead, Default Assignee, and Project administrator. These could be different people or the same person. The Project Lead is the person who manages the project. This could be, for example, the Project Manager or the Lead Developer. This role can be used in schemes.

When a user creates an issue, if an assignee isn't specified, the project's default assignee is used. By default, the issue is unassigned. You can configure the project so that the default assignee is the Project or Component Lead or unassigned. In Jira's general configuration, the setting 'Allow unassigned issues' is set to ON by default. Turn this off if you don't wish to allow unassigned issues across all projects.

The Project administrator can edit the project details, change the project type, and define project components, and versions. They can also view but not select or edit the project's schemes. Also an important responsibility is to manage project role membership. They also have limited editing of project workflows and screens, which we discuss on the next slide.

When you set Jira up, the jira-administrators group is a default member of the Administrators role, so unless a Project administrator changes this, Jira administrators are automatically Project administrators. And if they are not, they can change a project's permission scheme to make themselves Project administrators. Default schemes are configured to use the Administrators role.

## PROJECT ADMINS CAN

- Edit the workflows & screens associated with their project
- Create, update, and delete transitions
- Add existing statuses
- Remove statuses from the workflow
- Add, remove, and rearrange fields on a screen

## PROJECT ADMINS CAN'T

- Edit shared or system workflows & screens
- Edit properties, conditions, validators, or post functions
- Create new statuses
- Select or update a transition screen
- Create new custom fields



By default, project administrators get the Extended project administration permission. This permission allows them to edit their project's workflows and screens under certain circumstances.

The restrictions for editing their project's workflows:

- The workflow must not be shared with any other projects, or be a system workflow.
- To add a status, the status must already exist in the JIRA instance, which means that the project admin can't create new statuses or edit existing statuses.
- To delete a status, the status must not be used by any of the project's issues.
- The project admin can create, update, or delete transitions; they cannot select or update a screen used by the transition or edit or view a transition's properties, conditions, validators, or post-functions.

The restrictions for editing their project's screens:

- The screen must not be a default system screen.
- The screen must not be shared with any other projects or used as a transition screen in workflows.
- The project admin can add, remove, and rearrange system fields.
- The project admin can add, remove, and rearrange existing custom fields, but they cannot create custom fields.

# Creating & assigning members to custom roles



Jira administrator

1. Creates project role
2. Adds project role to scheme
3. Associates scheme with project



Project administrator

- Assigns project members to project role



In addition to the automatic roles created for every project, Jira administrators can create custom roles that apply across projects. You can use these roles in schemes that can be shared across projects. Each project can have unique role members.

To create and assign members to custom roles:

1. The Jira administrator creates custom project roles in Jira administration, so all projects have the same project roles available to them.
2. The Jira administrator adds project roles to schemes. The core usage for project roles is for permission and notification schemes. While you could assign permissions and notifications to users and groups directly, roles are more flexible and sustainable.
3. The Jira administrator then associates the scheme with a project.
4. Then Project administrators assign members to project roles specifically for their project(s). Of course, the Jira administrator can also assign project members to project roles, but ideally, they'd delegate this work to the project administrator.



# Permissions for Project Administrators

## Project permissions

Permission	Granted to
<b>Administer Projects</b> Ability to administer a project in JIRA. <input checked="" type="checkbox"/> Extended project administration Grant extended <a href="#">project administration permissions</a> .	<b>Project role</b> <ul style="list-style-type: none"><li>Administrators</li></ul>

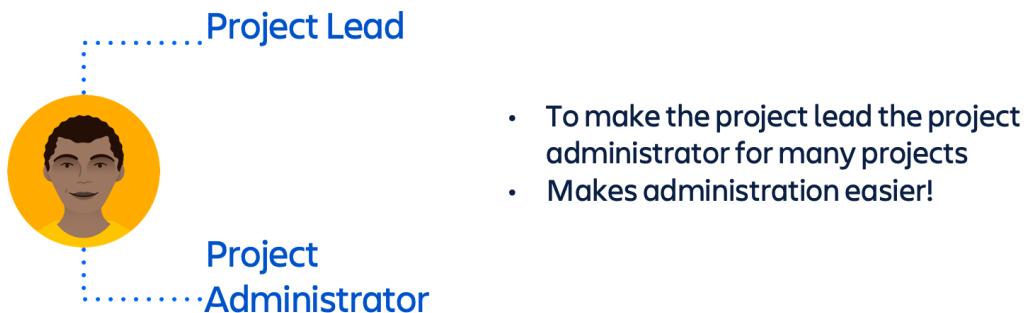
Required for project administrators to edit their workflow



The Extended project administration permission is granted by default to project administrators in the permission scheme. This permission enables project administrators to edit workflows and screens under certain limited circumstances.

You can deselect this option to disable the extended project administration permission for projects where Project administrators do not edit workflows or screens.

# Why Assign the Administer Projects Permission?



Assigning the Administer Projects permission to Project Leads is a way of reducing administration overhead. How? Well, the Project Lead typically performs the tasks of a Project administrator, such as assigning people to roles, updating project details, and so on. If this is the expectation in your organization, it makes sense to automatically grant Project Leads the Administer Projects permission.

You can set this up for the default permission scheme for all projects of an application type or all categories of projects, such as all development projects.

There are benefits to this configuration. The Project Leads for all projects using this permission scheme now automatically have the Administer Projects permission. You don't need to go into each project and grant a user the Administer Projects permission, which reduces administrative overhead. And if the Project Lead changes, the new Project lead automatically has the Administer Projects permission.

This configuration might not meet your requirements, however. Some organizations find that this grants too many users too many administrative permissions. For example, you might use the project lead for one purpose; so that they are the default assignee.

# Custom Roles



In addition to the automatic roles created for every project, you can create custom roles that apply across projects. You can use these roles in permission schemes that are shared across projects. Each project can have unique members for this role.

For example, if your organization requires all software development issues to be tested and then closed by a Quality Assurance (QA) person, you could do the following:

1. Create a project role called QA.
2. Create a permission scheme called Dev Permission scheme, in which you assign the 'Resolve Issues' permission only to the QA project role.
3. Associate the Dev permission scheme with all software development projects. In this example, the projects are Astronaut App, Rocket App and Music App
4. For each software development project, add the appropriate QA people to the QA project role. For example in the Astronaut App, Dave and Mary have been assigned this role.

# Creating & Assigning Members to Custom Roles



Jira administrator

1. Creates project role
2. Adds project role to scheme
3. Associates scheme with project



Project administrator

- Assigns project members to project role



All projects have the same project roles available to them. The Project administrators assign members to project roles specifically for their project(s).

In addition to the automatic roles created for every project, Jira administrators can create custom roles that apply across projects. You can use these roles in shared schemes. Each project can then have unique members for the role.

To create and assign members to custom roles:

1. The Jira administrator creates custom project roles in Jira administration. All projects have the same project roles available to them.
2. The Jira administrator adds project roles to schemes. The core usage for project roles is for permission and notification schemes. While you could assign permissions and notifications to users and groups directly, roles are more flexible and sustainable.
3. The Jira administrator then associates the scheme with a project.
4. Then Project administrators assign members to project roles specifically for their project(s). Of course, the Jira administrator can also assign project members to project roles, but ideally, they'd delegate this work to the project administrator.

## Scenario



Human Resources  
Project Administrator

I want **two consultants** on my HR project to be able to **create issues** in this project

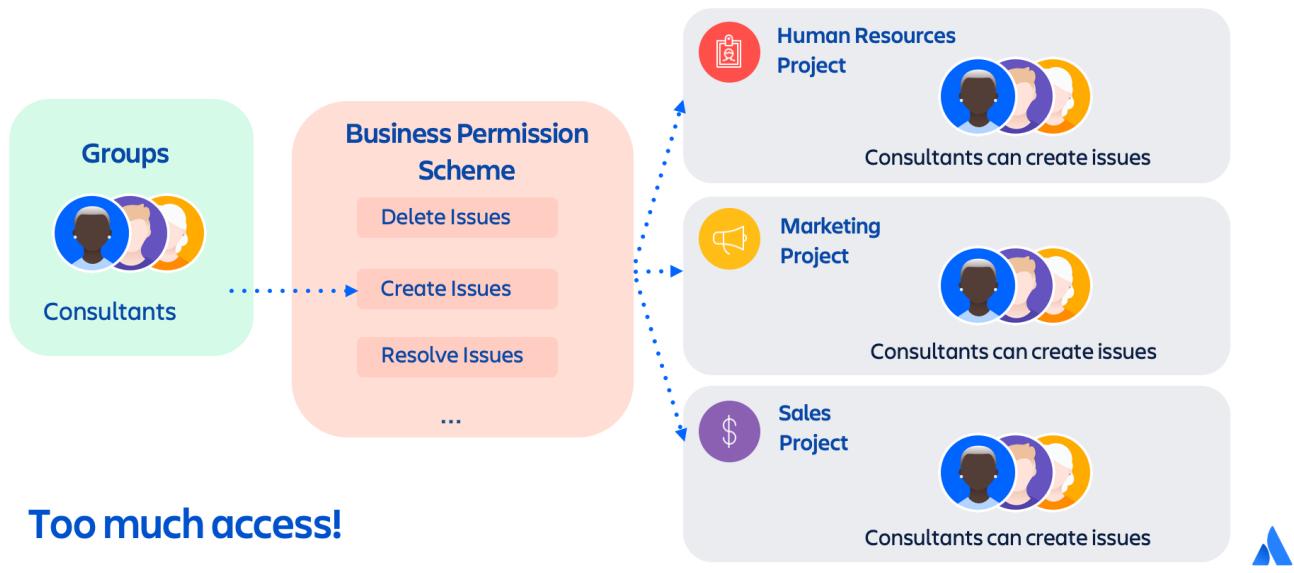
?

*What are our options?*



In this scenario, the project administrator for the Human Resources project wants two consultants on her project to be able to create issues in her project temporarily. The consultants need the Create Issues project permission. What are the options? We'll look at them in the next slide.

# Option 1 – Use group to update shared scheme



Option 1 – use a group to update a shared scheme:

1. The Jira administrator creates the Consultants group and adds all consultants for all projects to this group.
2. The Jira administrator updates the shared Business Permission Scheme and grants the Consultants group the Create Issues permission.

This option is an easy change for the Jira administrator, but is it a good option?

In this scenario, three Business projects are sharing the Business permission scheme; Human Resources, Marketing, and Sales.

The problem with using a group to update the shared scheme is that all members of the consultants group can create issues in all the Business projects that use the Business permission scheme. Each project uses a couple of consultants, and they're different for each project.

This solution gives too much access to the members of the Consultants group, which isn't a desirable option. Also, it involves two different people to make the change if two people perform the roles of Jira administrator and Project administrator.

## Option 2 – Use groups to update new schemes



Option 2 – use groups to update new schemes:

1. The Jira administrator creates the HR Consultants group and adds each consultant.
2. The Jira administrator then:
  - a. Copies the Default Permission Scheme to create a new HR Permission Scheme.
  - b. Updates the HR Permission Scheme and adds the HR Consultants group to the Create Issues permission.
  - c. Associates the HR Permission Scheme with the Human Resources project.

Then the Jira administrator goes through the same process for the Marketing project, creating the Marketing Consultants group and a Marketing Permission Scheme.

This approach solves the problem of one Consultants group having permissions in many projects. Is this a good option?

Since requirements are often very specific (different teams, stakeholders, and so on), this is likely to be a common requirement across projects. The Jira administrator would have to create many schemes and be involved in every requested change. Each project's scheme would need to be changed every time there's a requirement change, which causes a lot of administrative overhead. It's also time consuming and doesn't offer much flexibility. Also, having many, many schemes can potentially incur a performance hit.

Also, this solution requires two people to make the change. Maybe there's a better way ...

# The solution – Project roles in schemes



The solution is to use project roles. In this scenario the Jira administrator:

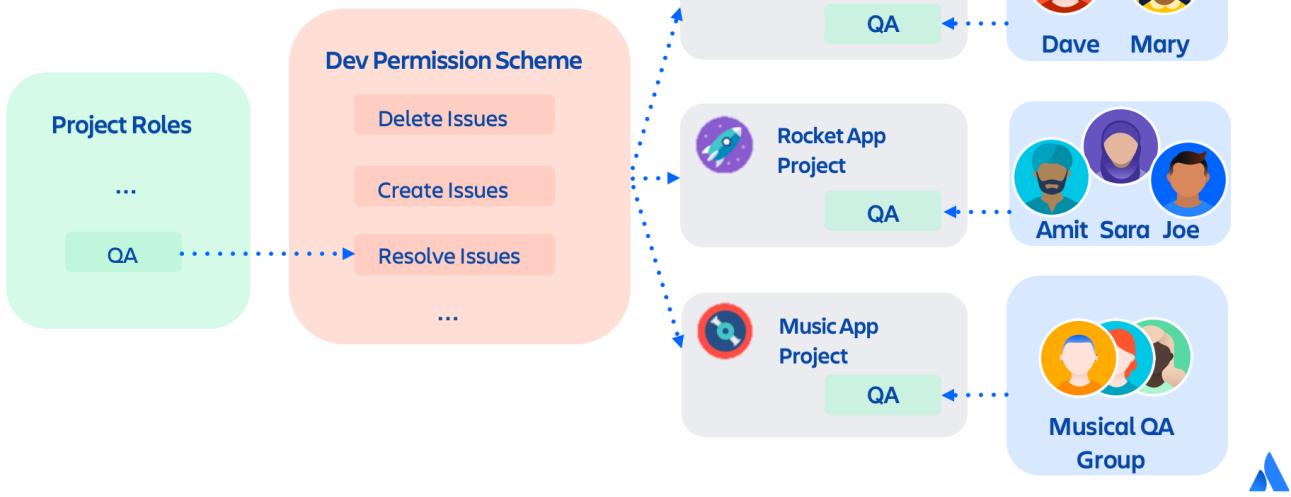
1. Creates the Consultants project role
2. Copies the Default Permission Scheme to create the Business Permission Scheme.
3. Adds the Consultants project role to the Create Issues permission in the scheme.
4. Associates the Business Permission Scheme with all projects that use consultants.

Finally, the project administrators add their consultants to the Consultants project role.

Initially, there's a bit of work for the Jira administrator to do, but it's a lot easier each time a request comes up. The Jira administrator needs to associate this permission scheme with the new project and let the project administrator define their project members. Also, only the Project administrator and Jira administrator are involved.

Here we see the power of roles and schemes. You use roles in schemes. Then you share the schemes among projects. By assigning the Create Issues permission to a project role, whoever has that role in a particular project, will have that permission. So in this example, consultants Ravi and Sue can create issues in the Human Resources project, Sam, Ava, and Nalini can create issues in the Marketing project, and the Sales Consultants group can create issues in the Sales project. By using project roles, you can use fewer permission schemes. Fewer schemes can mean better performance. It also means you can share schemes and save a lot of administration work. That means you can focus on the tasks that can't be delegated to project administrators.

## Another custom roles example

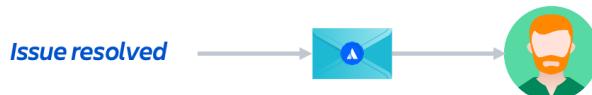


Here's another example of custom roles. If your organization requires all software development issues to be tested and then closed by a Quality Assurance (QA) person, you could do the following:

1. Create a project role called QA.
2. Create a permission scheme called Dev Permission scheme, in which you assign the 'Resolve Issues' permission only to the QA project role.
3. Associate the Dev permission scheme with all software development projects.
4. For each software development project, add the appropriate QA people to the QA project role. For example, in the Astronaut App, Dave and Mary have been assigned this role. Whereas in the Rocket App project, Amit, Sara, and Joe have been assigned to the QA role. And in the Music App project the Musical QA group has been assigned to the QA role.

# Using Roles to Manage Notifications

Jira can notify the users when a particular event occurs in your project



The easiest way to manage notifications is with project roles

If you have an outgoing (SMTP) mail server set up, email notifications will be sent as soon as there is any activity in the new project!



Jira can notify the appropriate people when a particular event occurs in your project, on events such as Issue Created or Issue Resolved. You can choose specific people, groups, or roles to receive email notifications when different events occur. As a best practice, roles are the easiest way to manage notifications.

- Notifications are managed in notification schemes. The project's notification scheme determines who receives email notifications of changes to issues in this project.
- Email settings specify the 'From' address for emails sent from this project. These settings are only available if an SMTP email server has been configured in Jira.

Please note, the default notification scheme for each project type is associated with all new projects of the corresponding type by default. If you have an outgoing (SMTP) mail server set up, email notifications will be sent as soon as there is any activity, such as when issues are created in the new project.

# Troubleshooting Permissions

The Permission Helper helps you diagnose why a user can or cannot see a certain issue

The screenshot shows the 'Permission helper' dialog box. At the top, there are three dropdown menus: 'User' set to 'agrant', 'Issue' set to 'HR-6 - Test 6', and 'Permission' set to 'Browse Projects'. Below these, a summary section displays the following details:

- Permission name: **Browse Projects**
- User: **Alana Grant**
- Project: **Human Resources**
- Permission scheme: **HR Permission Scheme**
- Issue: **HR-6**
- Status: **Alana Grant has the 'Browse Projects' permission**

Below this summary, there is a table with two rows:

Status	Summary	Details
✓	Jira Service Desk does not override this permission	Jira Service Desk does not override this permission
✓	Application Access	Any logged in user has this permission

At the bottom right of the dialog are 'Submit' and 'Close' buttons.



The Jira Permission Helper can help you diagnose why a user can or cannot see a certain issue.

1. Choose Permission schemes > Permission Helper.
2. Enter the username of the user (leave blank for anonymous users), an issue key (for example, an issue that the user can/cannot see) and the permission to check.
3. Click Submit.

# Control what users can do

**Users & Groups**

**Application Access**

**Global permissions**

**Permission Schemes**

**Project Roles**



Jira permissions revolve around users, groups, application access, global permissions, permission schemes, and project roles.

- Users need application access to log in to Jira and use the features of a particular Jira product - Jira Software, Jira Core, and Jira Service Management.
- Groups are logical groupings of users in your instance that can be used to control users' application and project-level permissions.
- Global permissions are system-wide and granted to groups of users. The global permissions are Jira Administrators, Browse Users, Create Shared Objects, Manage Group Filter Subscriptions, and Bulk Change. Jira Data Center has an additional global permission, Browse Archive.
- Project-level permissions are managed in permission schemes. Permission schemes grant project-level permissions to users, groups, or project roles. A Jira project uses one permission scheme. Permission schemes can be shared across projects.
- Project Roles are a flexible way to manage users at the project-level. Role membership can change from project to project, while group membership is managed at the application administration level.

## See how it's done



- View the default permission schemes for different types of projects
- Change permissions so only project leads can move issues in Software projects



In this demo we'll look at the default permission schemes. We'll also look at managing permissions.

# Are you getting it?



By default the Default Permission Scheme is used for all new Business type projects.

- a. True
- b. False



The answer is on the next slide.

## Did you get it?



By default the Default Permission Scheme is used for all new Business type projects.

- ✓ a. True
- b. False



### Answers:

1. a



## Are you getting it?

Use roles in permission and notification schemes to (select 3):

- a. Create fewer schemes
- b. Reduce administrative overhead
- c. Restrict project administrators from assigning users to project roles
- d. Avoid the use of users and groups in schemes



The answer is on the next slide.

## Did you get it?



Use roles in permission and notification schemes to (select 3):

- a. Create fewer schemes
- b. Reduce administrative overhead
- c. Restrict project administrators from assigning users to project roles
- d. Avoid the use of users and groups in schemes



Answer: a, b, and d

## Takeaways



- Assign permissions to project roles (not groups) in permission schemes
- Create schemes that are shared across similar projects rather than unique schemes for every project



As a best practice, use of project roles wherever possible to allow the use of fewer permission schemes. Having a unique set of schemes for every project causes administrative overhead and can result in Jira system degradation in larger installations. Keep your system lean and easier to manage by assigning permissions to project roles in permission schemes.

You can choose to grant project permissions to specific users, groups, or roles. A benefit of using roles is that their membership is managed at the project-level by Project administrators, which reduces application management overhead for the Jira administrator.

Use project roles in permissions schemes rather than users or groups to keep your schemes flexible and portable. Avoid using groups and users in permission schemes and do so only when for security reasons, you cannot delegate management to Project administrators.

Create schemes that can be shared across similar sorts of projects, such as projects that manage product development and have similar requirements rather than using unique schemes for every project.

# Try it

## Lab 6 - Configuring Project Roles & Permissions



- Exercise 1 – Assigning Project Roles
- Exercise 2 – Creating Project Roles & Configuring Project Permissions
- Optional Exercise 3 – Exploring Extended Project Administration
- Optional Exercise 4 – Viewing Other Role & Permission Information



In this lab, you'll look at configuring project roles and permissions.

# Do you want to be a Jira administrator superhero?



## Jira Server Administration Part 2: Taking it to the next level!



Do you want to be your organization's dream Jira administrator?

Take our next course in this series Jira Server Administration Part 2: Taking it to the next level! The goal of this course is to make you a Jira administrator superhero who can roll out a Jira instance that improves everyone's performance and happiness! You have the ability to shape and mold Jira to fit the way your teams work, to craft Jira to make your teams hum! In this course you'll learn how to :

- Map your organization's business requirements to Jira configurations
- Create issue types, fields, and screens to make it easy for your users to get their work done
- Customize your workflows to match how your teams work
- Configure sprint and board permissions so users have appropriate access
- Create a set of standard schemes and configurations that you can use to stamp out new projects and apply to already existing projects

## Earn Atlassian Certifications!



Propel your career



Boost your skills



Help your teams do  
their best work

[atlassian.com/certification](https://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](https://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](https://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?

<b>Training &amp; Certification</b>	<a href="https://atlassian.com/university">atlassian.com/university</a>
<b>Training &amp; Certification Online Community</b>	<a href="https://go.atlassian.com/traincertcommunity">go.atlassian.com/traincertcommunity</a>
<b>General Online Community</b>	<a href="https://community.atlassian.com">community.atlassian.com</a>
<b>Documentation</b>	<a href="https://confluence.atlassian.com">confluence.atlassian.com</a>
<b>Support</b>	<a href="https://support.atlassian.com">support.atlassian.com</a>
<b>Enterprise Services</b>	<a href="https://atlassian.com/enterprise/services">atlassian.com/enterprise/services</a>
<b>More Resources</b>	<a href="https://atlassian.com/resources">atlassian.com/resources</a>



# 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](https://aug.atlassian.com) to join a group near you.

## No ACE in your city?

You can start one! Go to [aug.atlassian.com/leaders](https://aug.atlassian.com/leaders) to sign up. You can also connect with fellow users on [community.atlassian.com](https://community.atlassian.com)



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



# Congratulations on completing the course!



A

# Appendix A: User Directories



In this appendix module, we'll look at user directories.

# What will you learn?

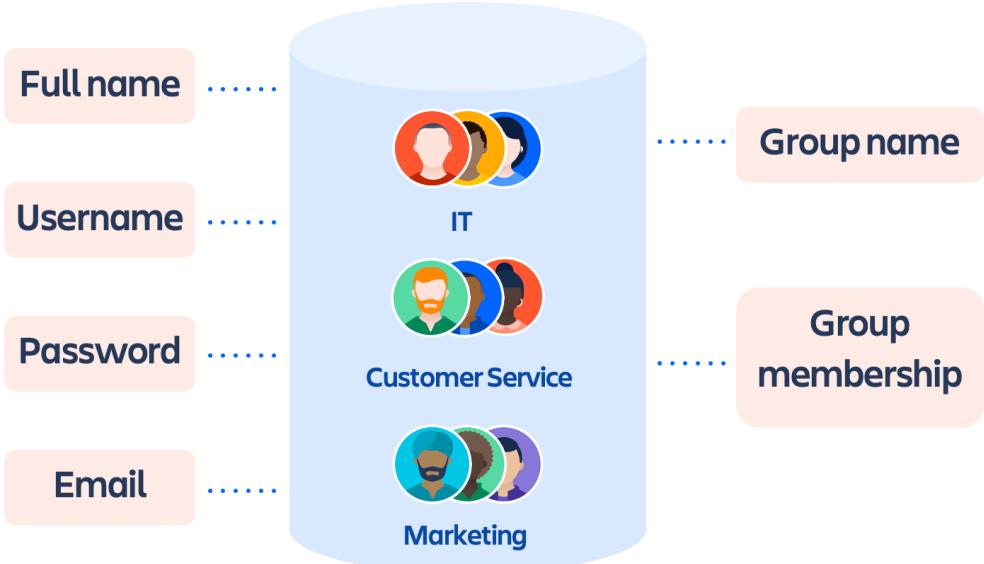


- Identify user directories
- Synchronize a user directory



We'll look at using external directories to manage users in Jira.

## User Directories



A user directory provides persistent storage for information about users and groups. User information includes the person's full name, username, password, email address, and other personal information. Group information includes the name of the group, the users that belong to the group, and possibly groups that belong to other groups.

# Where Does Jira Store User Data?



“We have 10,000 users and need to manage our Jira users in our company’s existing LDAP.”



There are many combinations of where user data may come from. Jira comes with an inbuilt internal user directory, and this may be sufficient to store your user data. You can also add external user directories. In this example, this organization has 10,000 users. It uses LDAP for its Corporate directory, Crowd for single sign-on (SSO), and the Jira internal directory for temporary guest users.

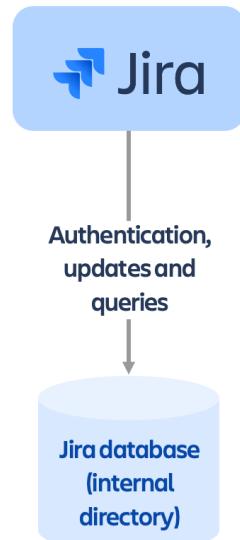
The following types of directory servers and directory managers are supported:

- An LDAP directory for delegated authentication. A wide range of LDAP providers is supported, such as Microsoft Active Directory or OpenLDAP. An LDAP directory is a collection of data about users and groups. LDAP (Lightweight Directory Access Protocol) is an Internet protocol that web applications can use to look up information about those users and groups from the LDAP server. You can connect your Jira application to an LDAP directory for authentication, user and group management. Connecting to an LDAP directory server is useful if your users and groups are stored in a corporate directory.
- Atlassian Crowd - Groups are managed fully (unless they sit on top of LDAP). Crowd can collate multiple directories. Note, Crowd is not covered in this course. Crowd is the only supported SSO right now. You can use other SSO solutions, but Atlassian doesn’t support them.
- You can also use another Jira server.

For Jira Data Center, Atlassian released the SAML in October 2016

## Jira Internal Directory

- User/group information stored in the Jira database
- Enabled by default at installation
- Admin user created here at installation
- Cannot be removed



Keep at least one Jira administrator in the Jira Internal Directory

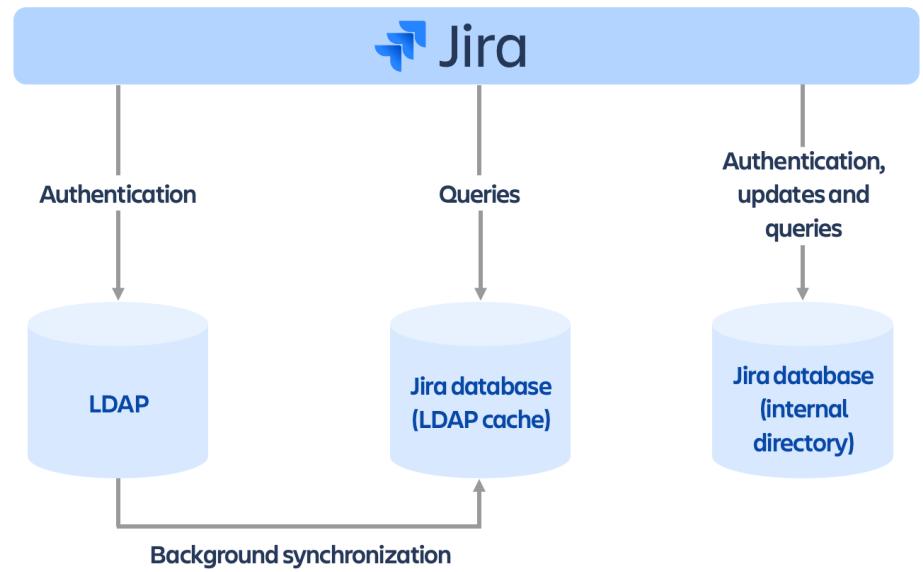


The Jira Internal Directory stores user and group information in the Jira database. It is enabled by default at installation.

When you create the first administrator during the setup procedure, that administrator's username and other details are stored in the internal directory.

You should maintain at least one Jira system administrator in the Jira Internal Directory. Even if your primary login is through an external directory (such as LDAP, Crowd, or Active Directory), this strategy prevents a lockout in the event something fails with the external directory. It also enables troubleshooting.

## LDAP - Read Only

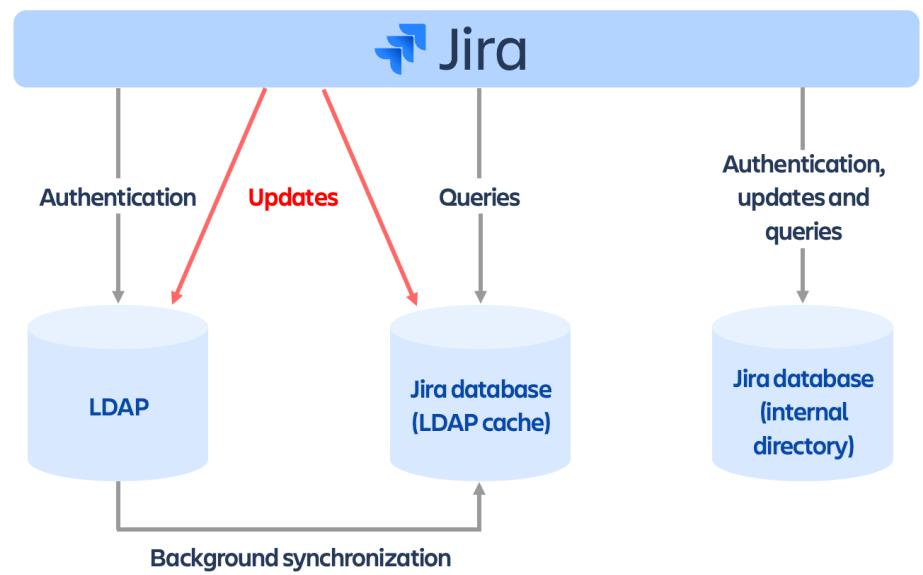


When configuring the directory, you can choose to make it read-only, read-only with local groups, or read/write. The purpose of this option is to identify whether changes you make in Jira can cause updates in the external directory.

Here we see a diagram for read-only (Read/Write and Read-Only with Local Groups are covered on the next slides). If you choose Read Only, Jira copies users and groups from the LDAP directory to the LDAP cache stored in the Jira database. Jira only reads from your LDAP directory; it is not allowed to change it. LDAP users, groups, and group memberships can only be modified via your LDAP Directory server. You cannot modify them from Jira Administration. This works well in environments where all identity management (groups included) is handled externally. A periodic synchronization task will run to update the Jira database LDAP cache with any changes made in the external directory. More on synchronization soon.

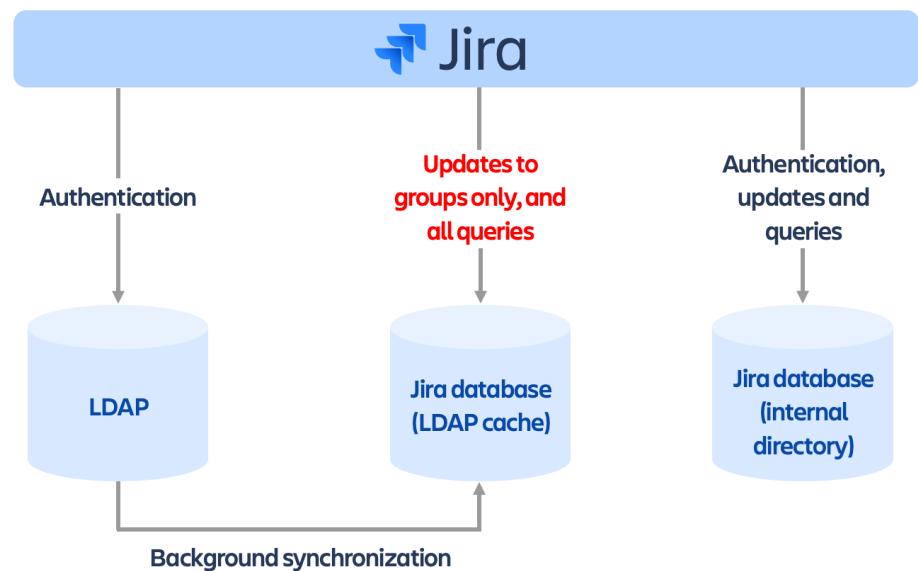
The Jira database does not store/cache user passwords for external user directories. All authentication happens via calls to the external directory.

## LDAP - Read/Write



If you choose Read/Write for the LDAP user directory, Jira copies users and groups from the LDAP directory to the LDAP cache stored in the Jira database. But any changes made to user and group information in Jira will also update the external LDAP directory. This option is rarely used. Usually, the Jira administrator and the LDAP administrator are different people. When LDAP is set up for many organizations, this is the single source of truth and should not be updated from Jira.

## Read Only, With Local Groups LDAP



Using Read Only with Local Groups for your LDAP user directory is similar to Read Only, with one additional feature; you can add LDAP users to local groups created in the Jira Internal Directory. You will go through this scenario in the lab.

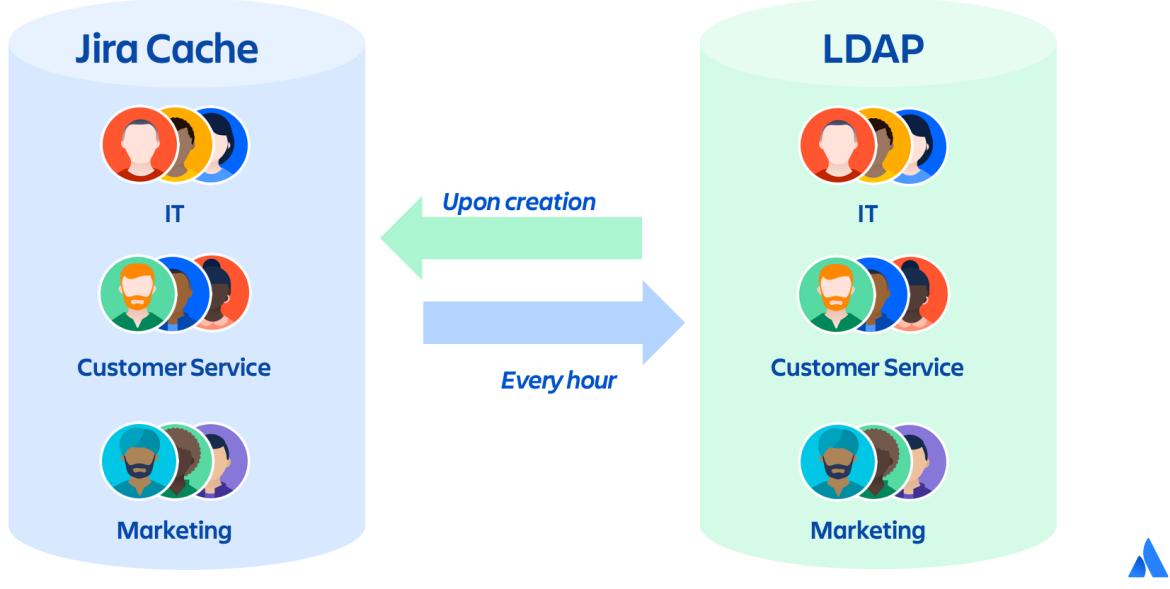
See

<https://confluence.atlassian.com/display/AdminJIRAServer085/Diagrams+of+possible+configurations+for+user+management>.

There are two other configurations with LDAP that we will not cover here:

- Jira connecting to LDAP directory for authentication only.
- Jira connecting to an LDAP directory for authentication only, with each user copied to the internal directory when they first log in to Jira.

# LDAP Synchronization



When you connect a new external user directory to the application, a synchronization task will start running in the background to copy all the required users, groups, and membership information from the external directory to the LDAP cache in the application database. For example, if you have a Marketing group with users in LDAP, when you create an LDAP user directory, a Marketing group with the same users will be cached in Jira. This caching of external directories aids performance. The caches are held in the application database, and no user passwords are stored. All authentication happens via calls to the external directory. If the external directory permissions are set to read/write, whenever an update is made to the users, groups, or membership info via the application, the update will also be applied to the cache and the external directory immediately.

The default synchronization/polling interval is 1 hour. You can also manually synchronize if you make changes to the directory. You can tune the interval depending on:

- The length of time you can tolerate stale data.
- The amount of load you want to put on the application and the directory server.
- The size of your user base.

If you synchronize more frequently, your data will be more up to date. However, you may overload your server with requests. For recommendations on the optimal number of users and groups in your LDAP directory to not affect performance during LDAP synchronization, see <https://confluence.atlassian.com/display/AdminJIRAServer085/User+management+limitations+and+recommendations>.

# User Directory Order



 Avoid duplicate usernames across directories!



The order of the directories is the order in which they will be searched for user and group information and for authentication.

The order of user directories in Jira is important because it identifies which account "wins" when there is a conflict. For example, if you had an internal Jira directory user "Niko" and an LDAP Directory user "Niko" and he had a different password in each directory. When Niko logs in, the order of directories would determine which password he needed to authenticate. The one listed first is the one that Niko would need to log in as.

Tip: Users should only exist in a single directory. Avoid duplicate usernames across directories!

Also, changes to users and groups will be made only in the first directory, where the application has permission to make changes.

# Disabling & Removing User Directories

- You have to disable a directory before you can remove it.
- If you disable a directory, Jira won't recognize the users and groups in that directory.
- Removing a directory removes the details from the application database.
- You can't disable or remove the directory the Administrator belongs to.

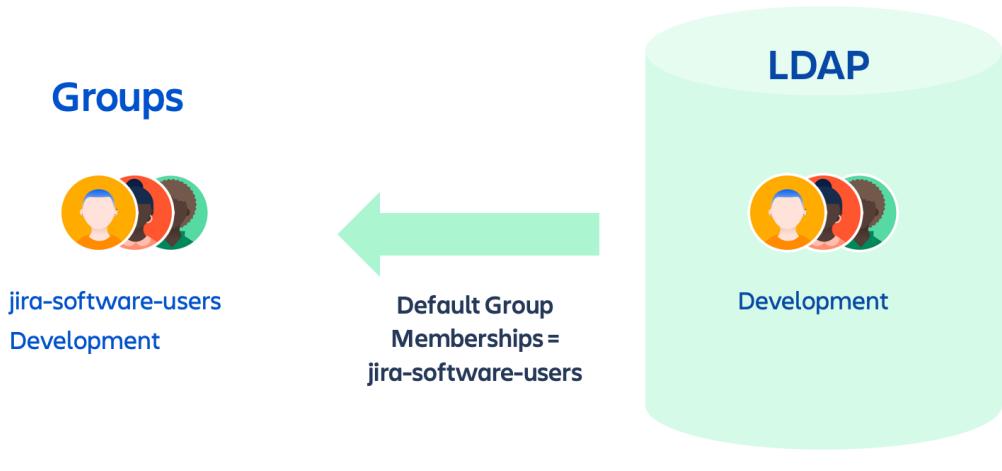


You can enable or disable a directory at any time. If you disable a directory, your configuration details will remain, but the application will not recognize the users and groups in that directory.

You have to disable a directory before you can remove it. Removing a directory will remove the details from the Jira application database.

You can't disable or remove the directory the Administrator belongs to.

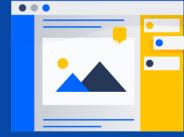
# Adding LDAP Users to Default Groups



When you create an LDAP user directory, all LDAP users and groups are automatically synchronized to the Jira LDAP cache in the Jira internal database. But what about membership in the application default groups which reside in the Jira internal directory? You can specify one or more Jira groups in the 'Default Group Memberships' setting when you create the LDAP user repository. This means all LDAP users will automatically be added to these groups. For example, if you have Jira Software installed. When you create your LDAP user repository, you enter `jira-software-users` in Default Group Memberships. So all LDAP users will automatically be added to `jira-software-users` and will have access to Jira Software.

It gets more complicated when you have multiple Jira applications installed. All LDAP users are added to all of the groups that you specify in default group memberships for each application. So if you have Jira Software, Core, and Service Management installed, you should divide your users up into the application default groups – `jira-software-users`, `jira-core-users`, and `jira-servicedesk-users`. If you specify `jira-software-users`, `jira-core-users` and `jira-servicedesk-users` in Default Group Memberships, all your LDAP users will be added to all three groups. It may be easier, in that case, to leave Default Group Memberships blank and use your existing LDAP groups e.g. Marketing, Customer Service, etc. to do bulk adds to default groups in Jira. We will cover this scenario in the lab.

## See how it's done



- Explore the user directories set up on your lab machines
- View LDAP user directory settings
- View LDAP troubleshooting information



In this demo, we'll look at the user directories on your lab environments. We'll view the LDAP user directory settings and view LDAP troubleshooting information.

# Are you getting it?



What happens when you disable a directory?

- a. Added users are removed from the directory
- b. The application will not recognize the users and groups in the directory
- c. The user data will be removed from the database
- d. All of the above



Answer is on the next slide.

## Did you get it?



What happens when you disable a directory?

- a. Added users are removed from the directory
- b. The application will not recognize the users and groups in the directory
- c. The user data will be removed from the database
- d. All of the above



Answer: b

# Are you getting it?



Where's the best place to store the admin user?

- a. Jira Internal Directory
- b. External directory such as LDAP
- c. Both an internal and an external directory



Answer is on the next slide.

## Did you get it?



Where's the best place to store the admin user?

- ✓ a. Jira Internal Directory
- b. External directory such as LDAP
- c. Both an internal and an external directory



Answer: a Always store the admin user in the internal directory.



## Takeaways

- Log in as an internal user when making changes to an external directory configuration.
- Keep an administrator in your Jira Internal Directory.
- Users should only exist in one directory.



Here are some best practices for working with user directories.

The recommended way to edit directory configurations is to log in as an internal user when making external directory configurations.

You should maintain at least one user that's both in the Jira Internal Directory and is a Jira system administrator even if your primary login is through an external directory such as LDAP, Crowd, or Active Directory. This strategy prevents a lockout in the event something fails with the external directory. We recommend that you keep either an administrator or system administrator user active in your internal directory for troubleshooting problems with your user directories.

You cannot edit, disable, or remove the directory to which your user, the administrator, belongs. This precaution prevents administrators from locking themselves out of the application. It changes the directory configuration in a way that prevents administrators from removing their administration permissions.

We recommend that users only exist in a single directory and avoid duplicate usernames across directories.

# Try it

## Appendix A – Exploring User Directories

- Exercise 1 – Exploring & Synchronizing User Directories



In this lab, you'll explore user directories.