## **A ATLASSIAN** University

# Managing Jira Projects Cloud

Lab Workbook

# **A ATLASSIAN** University

### **Table of Contents**

Introduction	1
Lab format	1
Lab 1 Course Overview	3
Lab 2 Managing Projects	5
Exercise 1 Creating a Project	
Appendix	
Lab 3 Managing Roles & Permissions	10
Required for the lab	10
Exercise 1 Managing Roles & Permissions	12
Appendix	
Lab 4 Managing Boards	17
Required for the lab	17
Exercise 1 Adding a New Column	19
Exercise 2 Modifying the Board Filter	21
Exercise 3 Managing Releases and Versions	23
Appendix	26
Lab 5 Boards & Projects	28
Required for the lab	28
Exercise 1 Creating Multiple Boards in a Project	29
Exercise 2 Showing Multiple Projects on a Board	
Appendix	36
Lab 6 Managing Issues	38
Required for the lab	38
Exercise 1 Configuring Issue Types	39
Exercise 2 Configuring Components	41
Exercise 3 Using Bulk Change to Correct Labels	43

Exercise 4 Using Bulk Change to Move Issues from One Board to Another	44
Appendix	
Lab 7 Automation	
Exercise 1 Creating an Automation Rule	51
Appendix	56
Lab 8 Reports & Dashboards	57
Exercise 1 Creating a Dashboard	58
Exercise 2 Running Reports	62
Appendix	64
Lab 9 Other Jira Features	66
Required for the lab	66
Exercise 1 Creating a Project with Shared Settings	67
Exercise 2 Running Parallel Sprints	69
Appendix	70
Lab 10 Creating & Configuring Team-Managed Projects	71
Exercise 1 Creating a Team-managed Project & Controlling Access	72
Exercise 2 Adding & Configuring Issue Types	74
Exercise 3 Configuring the Board & Customizing Features	76
Appendix	77

### Introduction

### Lab format

### Optional exercises and appendices

The labs may have optional exercises. These are not required to complete the course. However, if you have time and interest, they supplement the exercises for the lab. There may also be appendices that you don't need to complete the class. They are full of useful information like additional reading and best practices. Dig into these after you complete the course!

### Language and User Interface

The language you see in the Atlassian product UI is set to your browser's language. If you wish to see the UI in English (to match the lab instructions), or in a different language, go to your Atlassian user profile and edit your account preferences.

Cloud products are constantly being updated with new features, so you may see some slight differences between the lab instructions and the product you are using.

### Logging in to your lab environment

To log in during the labs, you need your assigned site URL and the user's email address specified in the lab directions and password. If you're taking an On Demand course, you'll find these in the Virtual Lab Instructions activity in the Lessons section. If you're taking an instructor-led course, your instructor will share these details with you.

i The password for every user is the same. Keep this password easily accessible.



When switching between products in these labs, you can see other sites. It's important that you choose the product on the site that's been assigned to you.

Here's a list of the user(s) and what role they have in this course.

Name	Role
Dakota Jones	Jira administrator
Max Taylor	Project administrator
Sophie Nguyen	Development project member

Luis Beck	Development project member

### Lab 1 Course Overview

#### Logging into your lab

To log in during the labs, you need an assigned **site URL** and each user's **email address** and **password**. If you're taking this as an OnDemand course, you'll find this in the **"Virtual Lab Instructions"** activity. If you're taking an instructor-led course, you'll receive the details from your instructor.

The password for every user is the same.

#### Do not log in with your own Atlassian ID

You probably already have an Atlassian account that you use to log in to your own Atlassian products. In the labs for this course, a specific set of users has been added to the cloud site. You will log in with these accounts. Do not log in using your own Atlassian ID.

#### Log in using a new browser, or an Incognito or Private window

A single browser can only handle one Atlassian account. This is because browsers keep cookies. Once you're logged into a cloud site on one browser, it remembers that login. So, if you open a new tab, you can't login as someone else.

To log into your lab, use a different browser to the one you usually use, or use an incognito or private window to log in.

#### Logging into labs as different Users

In the labs you'll need to log in as one or more users. To avoid logging in and out a lot you can either use different browsers or use an incognito or private window for each user.

#### Opening an Incognito window

i You can open either an incognito window (Chrome) or private window (Firefox) from the browser menu. Other browsers also have the same functionality.

#### Chrome

- 1. Either:
  - a. Click the Chrome three-dot (ellipses) menu button or
  - b. From the Chrome browser menu click File.
- 2. In the dropdown menu, click New Incognito Window

#### **Firefox**

- 1. Either:
  - a. Click the three-line Firefox application menu or
  - b. From the Firefox browser menu click File.
- 2. In the dropdown menu, click New Private Window

#### **Accessing your site**

1. Use your assigned site URL to navigate to your site.



### ✓ You're all set!

When you get to Jira/Confluence, you'll be told who to log in as.

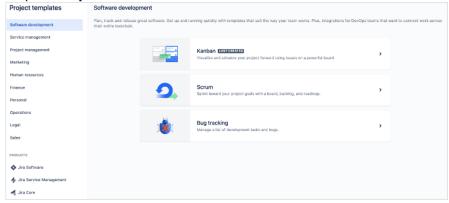
### Lab 2 Managing Projects

#### Scenario:

Your team is starting a new project to develop a smartphone app that tracks giant storms. To manage this development effort, you need a new Jira project. You log in as the Jira administrator to create the project as they're the only ones who can create projects.

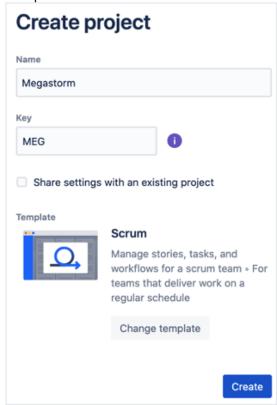
### Exercise 1 Creating a Project

- ① Only Jira administrators can create projects so we will be logging in as a Jira administrator. Once a project is created, Project administrators can maintain and configure the projects.
  - 1. Log in as the Jira administrator:
    - a. Go to your site URL and log in as the Jira administrator, **Dakota Jones**. Use the email address that corresponds to her name and the password that you received.
      - **Note:** It's important you log into the site that's been assigned to you. Do not log in with Google or Microsoft.
    - b. If you see onboarding screens, skip the questions or select whatever options you like
    - c. Once you're logged into Jira, bookmark the page. This will save you time later when logging in as other users.
- 2. Create a company-managed project:
  - a. If you don't land on the **Projects** page, go to it by clicking **Projects** in the top navigation bar and selecting **View all projects**.
  - b. On the Projects page, click **Create project** in the top right.
  - c. If you see the **Project templates** screen, **follow the instructions in this step**, then skip to **step 3**.



- i. Make sure **Software development** is selected in the sidebar.
- ii. Select Scrum.
- iii. Click **Use template**.
- iv. Click Select a company-managed project.
- v. Name the project **Megastorm** and leave the key at **MEG**.
- vi. Leave Share settings with an existing project unchecked.
- vii. Click **Create project** (if you see **Next** instead, select that, and on the following screen, choose **Skip**.) You should see the Kanban board for your new project.
- d. If you did not see the **Project templates** screen, follow these instructions:
  - i. Name the project **Megastorm** and leave the key at **MEG**.

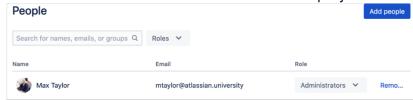
- ii. Leave **Share settings with an existing project** unchecked.
- iii. Click **Change template**.
- iv. Click the drop-down at the top of the page. Here you can filter project templates based on product. If it's not already selected, select **Software**.
- v. Here you see all the project templates for Jira Software. Ensure the **Scrum** template is selected.



#### vi. Click Create.

- Note: The key is used as the prefix of this project's issue keys. For
  example, your first issue's key will be MEG-1. If there are multiple words
  in the project name, the key will be filled in with the first letter of each
  of the words in the name.
- **Note:** You can change the default project key that's created for you, but if you do, choose a key that's descriptive and easy to type. Users will often use the issue key to find an issue and you want to make it as easy as possible for them so don't use overly long keys. The key can be changed after project creation, but it's better to start off with a suitable
- **Note:** The Share settings with an existing project setting lets you share project configurations such as issue types, workflow, and more with other projects. If a change is made to one of the projects' configurations, that change affects all the projects that share that configuration.

- 3. Assign the project administrator:
  - a. At the bottom of the project sidebar, click **Project settings** (or the cog icon).
  - b. In the project settings sidebar, click **People**.
  - c. On the People page, click **Add people** then in the pop-up:
    - i. Start typing max and select Max Taylor.
    - ii. For role, select Administrators.
    - iii. Click Add 1 person.
  - d. Max is now listed as an administrator in the project.



• **Note:** Even though Jira administrators are not listed on this page, they automatically have this role in all projects.

Congratulations on completing the lab!

### **Appendix**

### **Further Reading**

Reference	URL
Create a new project	http://go.atlassian.com/create_project_cloud

### **Best Practices**

Pitfall	Example Use Case	Best Practice
Users are having problems searching for issues in your project.	You changed the default project key from DEV to DEVLMNT.	If you change the default project key, choose a key that is descriptive and easy to type. Users will often use the issue key to find an issue and you want to make it as easy as possible for them so don't use overly long keys.

### Lab 3 Managing Roles & Permissions

#### Scenario:

The project administrator for the Megastorm project wants only the project administrator and the scrum master to manage sprints in the project. Also he wants the project administrator, the scrum master and one other team member to be able to delete issues in the project. So, the Jira administrator created two custom project roles and a new permission scheme to meet these specific business requirements. That way projects with similar requirements can use the new scheme and roles. In this lab, as the Jira administrator, you change the permission scheme for the project. Then, as the project administrator, you assign users to various project roles.

### Required for the lab

- 1. If you completed the second lab in this course, you can use the existing Megastorm project and skip to Exercise 1. If you didn't create this project, you need to create a new one:
  - a. Log in to your site as the Jira administrator, **Dakota Jones**.
  - b. On the Projects page, click **Projects** in the top navigation bar and select **Create** project.
  - c. If you see the **Project templates** screen, **follow the instructions in this step**, then skip to **step d**.



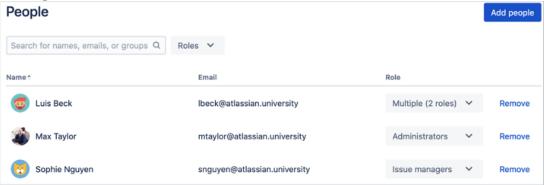
- i. Make sure **Software development** is selected in the sidebar.
- ii. Select **Scrum**.
- iii. Click **Use template**.
- iv. Click Select a company-managed project.
- v. Name the project **Megastorm** and leave the key at **MEG**.
- vi. Leave **Share settings with an existing project** unchecked.
- vii. Click Create project (if you see Next instead, select that, and on the following screen, choose **Skip**.)
- d. If you did not see the **Project templates** screen, follow these instructions:

- i. Name the project **Megastorm** and leave the key at **MEG**.
- ii. Leave **Share settings with an existing project** unchecked.
- iii. Click **Change template**.
- iv. Click the drop-down at the top of the page. Here you can filter project templates based on product. If it's not already selected, select **Software**.
- v. Here you see all the project templates for Jira Software. Ensure the **Scrum** template is selected.
- vi. Click **Create**.
- e. At the bottom of the project sidebar, click **Project settings** (or the cog icon).
- f. In the project settings sidebar, click **People**.
- g. On the People page, click Add people.
  - i. Start typing **max** and select **Max Taylor**.
  - ii. For role, select Administrators.
  - iii. Click Add.

### Exercise 1 Managing Roles & Permissions

- 1. Change permissions for a project:
  - a. Go to your site URL and log in as the Jira administrator, Dakota Jones.
  - b. Go to **Megastorm** project.
  - c. In the project sidebar, click **Project settings**.
  - d. In the project settings sidebar, click **Permissions**.
    - Note: This project uses the Default software scheme which all new Software projects use by default. A permission scheme associates permissions with users, groups or project roles.
       Many permissions use Application access (Any logged in user). This is any
      - Many permissions use Application access (Any logged in user). This is any logged in user who has access to Jira Software. Jira also adds the atlassian-addons-project-access project role to each permission. This role enables products and apps that extend Jira to have permissions to do their work.
  - e. View the **Manage sprints** permission.
    - Question: Who can manage sprints in this project?
    - **Answer**: Any user who has access to Jira Software and is logged in can manage sprints. (As well as the atlassian-addons-project-access role.)
  - f. View the **Delete Issues** permission.
    - Question: Who can delete issues in this project?
    - **Answer**: Project administrators can delete issues. (As well as the atlassian-addons-project-access role.)
  - q. At the top of the page, click **Actions** and select **Use a different scheme**.
  - h. Select **Development Permission Scheme** and click **Associate**.
    - **Question:** Now who can manage sprints in this project?
    - **Answer**: The project administrator and members of the Scrum masters project role can manage sprints. (As well as the atlassian-addons-project-access role.)
    - Question: Now who can delete issues in this project?
    - **Answer**: The project administrator and members of the Issue managers project role can delete issues. (As well as the atlassian-addons-project-access role.)
- 2. View project permissions as a project administrator:
  - In these labs you'll log in and out as various users. To lessen the amount of logging in and out, use different browser windows. However, you can't just open a new tab and log in as another user, as your original tab will be logged in as that user too. You have a few options, use the method that works best for you:
    - a. Open an incognito/private window in your current browser.
    - b. Open a separate browser if you have another one installed. For example, if you're using Google Chrome, open Firefox.
    - c. Log out as the current user by clicking their avatar. This method will require a lot of logging in and out!
    - d. Log in to your site as the Megastorm project administrator, Max Taylor.

- e. Go to the **Megastorm** project.
- f. In the project sidebar, click **Project settings**.
- g. In the project settings sidebar, click **Permissions**.
  - **Note:** As a project administrator (and not a Jira administrator), you can view your project's permissions, but you cannot change them. Note there is no Actions drop-down. Only Jira administrators can edit permission schemes.
- 3. Assign team members to project roles:
  - a. Assign Scrum masters and Issue managers custom project roles: In this project, Luis Beck is the scrum master. Max is going to assign Luis to the Project Lead, board administrator, and Scrum masters roles in the project. He will also assign Luis and Sophie Nguyen to the Issue managers role.
    - i. In the project settings sidebar, click **People**.
    - ii. Click Add people.
    - iii. Start typing luis and select Luis Beck.
    - iv. For Role, select Scrum masters.
    - v. Click Add 1 person.
    - vi. Repeat these steps to add **Luis Beck** and **Sophie Nguyen** to the **Issue** managers role.



- b. Assign project lead:
  - i. In the project settings sidebar, click **Details**.
  - ii. For Project lead, start typing **luis** and select **Luis Beck**.
  - iii. Leave the Default Assignee as Unassigned.
  - iv. Click Save details.
- c. Assign board administrator:
  - i. At the top of the project settings sidebar, click **Back to project**.
  - ii. On the Active sprints page, click the more (...) icon above the board and select Configure board (if the Enhanced board and backlog is enabled) or Board settings.
  - iii. Go to the **General** page (if you're not already there).
    - Question: Who is currently listed in Administrators and why?
    - **Answer**: Dakota Jones is the board administrator because she created the board (it was automatically created as part of creating a Software

Scrum project). Even though Max is not listed, as the project administrator, he can also configure the board.

- iv. Beneath Administrators, mouse over her name and click the edit icon.
- v. Start typing **luis** and select **Luis Beck**.
- vi. Click outside the field to exit edit mode.
  - Note: Now Luis, Dakota, and Max can configure the project's default board.
- 4. Verify your changes as the scrum master:
  - a. Log in to your site as **Luis Beck**.
  - b. Go to the **Megastorm** project.
  - c. In the project sidebar, click **Backlog**.
  - d. On the Backlog page, create two issues (of any type) e.g. Test 1 and Test 2.
  - e. Click **Create sprint**.
    - i. Drag the two issues up into the sprint and click **Start sprint**.
    - ii. Accept the defaults and click **Start**.
      - **Note**: Now you see the sprint board with the two issues. This confirms Luis has the Manage sprints permission.
  - f. Open one of the issues, then:
    - i. Click the more (...) icon in the top right.
    - ii. Confirm you see the **Delete** option (but don't delete it).
      - Note: This confirms Luis has the Delete Issues permission.
    - iii. Close the issue.
  - g. Back on the sprint board, click **the more (...) icon** and select **Configure board** or **Board settings**.
    - Note: This confirms he is a board administrator for the project's default board.
  - h. Log out as **Luis**.
- 5. Verify your changes as a developer:
  - a. Log in to your site as Sophie Nguyen.
  - b. Go to the **Megastorm** project.
  - c. On the Active sprints page, complete the sprint.
    - Question: Was Sophie able to complete the sprint? If not, why?
    - **Answer**: No, Sophie cannot complete the sprint, that option is not present. This is because she doesn't have the Manage Sprints permission.
  - d. Open one of the issues and confirm she has the option to delete the issue (but don't delete it).
    - Note: This confirms Sophie has the Delete Issues permission.
  - e. Back on the sprint board, click the three dots (...) and select **Configure board** or **Board settings**.
    - **Question:** Can she configure the board? For example, on the General page, can she edit the Administrators? If not, why?
    - **Answer**: No, Sophie cannot configure the board as she's not a board administrator. Note the message at the top telling her to contact a Jira or board administrator to configure this board.
  - f. Log out as Sophie.

N/	lanc	aina	lira	Proi	iects	Cloud
ıv	IUIIC	ıuııu	JII U	FIU	ICCL3	Cloud

Congratulations on completing the lab!

### **Appendix**

### **Further Reading**

Reference	URL
Managing project permissions	https://support.atlassian.com/jira-cloud- administration/docs/manage-project- permissions/
Managing project roles	https://support.atlassian.com/jira-cloud-administration/docs/manage-project-roles/

### **Best Practices**

Pitfall	Example Use Case	Best Practice
Users are not assigned the correct permissions in your projects.	The developers in your project complain they cannot see the development tools in their issues. It turns out you didn't assign your team members to the correct project role so they got the View Development Tools permission.	Familiarize yourself with the permissions set for your project and which roles get which permissions. Then assign members of your project team to the appropriate roles.

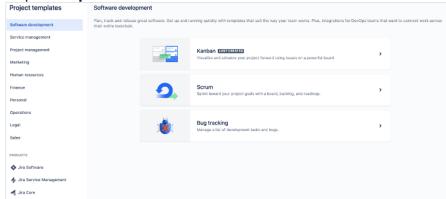
### Lab 4 Managing Boards

Scenario:

In these exercises, as the project administrator, you improve the usability of your scrum board by adding a new column and modifying the board filter. You also manage releases and versions in a project.

### Required for the lab

- If you completed previous labs in this course, you can use the existing Megastorm project. If you didn't create this project, you need to create a new one:
  - a. Log in to your site as the Jira administrator, **Dakota Jones**.
  - b. On the Projects page, click **Projects** in the top navigation bar and select **Create** project (if you see Next instead, select that, and on the following screen, choose
  - c. If you see the **Project templates** screen, **follow the instructions in this step**, then skip to **step d**.

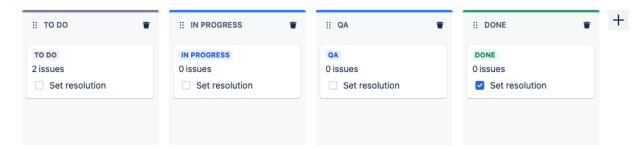


- i. Make sure **Software development** is selected in the sidebar.
- ii. Select **Scrum**.
- iii. Click **Use template**.
- iv. Click Select a company-managed project.
- v. Name the project **Megastorm** and leave the key at **MEG**.
- vi. Leave **Share settings with an existing project** unchecked.
- vii. Click Create project.
- d. If you did not see the **Project templates** screen, follow these instructions:
  - i. Name the project **Megastorm** and leave the key at **MEG**.
  - ii. Leave Share settings with an existing project unchecked.
  - iii. Click Change template.
  - iv. Click the drop-down at the top of the page. Here you can filter project templates based on product. If it's not already selected, select **Software**.

- v. Here you see all the project templates for Jira Software. Ensure the **Scrum** template is selected.
- vi. Click **Create**.
- e. At the bottom of the project sidebar, click **Project settings** (or the cog icon).
- f. In the project settings sidebar, click **People**.
- g. On the People page, click **Add people**.
  - i. Start typing max and select Max Taylor.
  - ii. For role, select Administrators.
  - iii. Click Add.

### Exercise 1 Adding a New Column

- (i) Your team has decided that all issues should go through a formal testing step before advancing to the Done column. Here you add a new column to the board, as well as a new status, called QA.
  - 1. Log in to your site as the project administrator, Max Taylor.
- 2. Go to the **Megastorm** project's **Active sprints** page.
- 3. Click the three dots (...) above the board and select **Configure board** (if the Enhanced board and backlog is enabled) or **Board settings**.
- 4. On the board configuration sidebar, click Columns.
- 5. Click Add column (or the + button):
  - a. Name: QA
  - b. Category: In Progress
  - c. Click Add column.
    - Note: The status QA was automatically created and mapped to the column.
       By adding a new status, as a result of adding a new column, you are modifying the underlying workflow for the board.
- 6. Grab your new **QA** column and move it before In Progress, then move it back to be between In Progress and Done.



- 7. Click **Back to board** and view your new QA column.
- 8. If you don't have any issues on the board, create a new issue in the project and add it to the sprint by doing the following:
  - a. Go to the **Backlog** page.
  - b. Click Create issue.
  - c. Select the issue type, enter the Summary, and hit the return key on your keyboard.
  - d. If necessary, create a sprint.
  - e. Drag the issue from the backlog into the sprint and, if prompted, click Confirm.
  - f. If necessary, start the sprint and click Start.
- 9. Open one of your issues on the board.
- 10. In the issue, click the status dropdown and select **View workflow**.

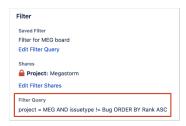


- **Note:** This is the workflow for issues on your board. It's using a simplified workflow. Note the All next to each workflow status, which means an issue can move from any status to any other status on the board. You see your new QA status. The statuses may look slightly different in color to what you see here.
- 11. Close the workflow dialog and the issue.

### Exercise 2 Modifying the Board Filter

- ① Your team is now focused on adding new features, but the board also shows bugs.

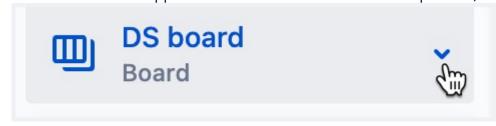
  They've asked to just hide bugs permanently. This can be done by modifying the board filter.
  - Only the owner of a filter can edit it. Since the Jira administrator created the project, and thereby the board, they own the shared filter for the default board. To allow the project administrator to edit the board filter, the Jira administrator will first change the ownership of the filter.
  - 1. Change the owner of the board filter:
    - a. Go to your site URL and log in as the Jira administrator, **Dakota Jones**.
    - b. Go to the **Shared filters** Jira administration page by either:
      - i. Type a . (period) and start typing **shared filters** and select **Filters Shared** items.
      - ii. Click the cog icon (Settings) in the menu and select **System**. In the sidebar, under SHARED ITEMS, click **Filters**.
    - c. In the Filter for MEG board row, click ... (three dots) and select Change Owner:
      - i. Start typing **max** and select **Max Taylor**.
      - ii. Click **Change Owner**.
    - d. Max should now be listed as the owner of the Filter for MEG board.
- 2. Modify the board filter:
  - a. Log in to your site as the project administrator, Max Taylor.
  - b. Go to the **Megastorm** project's **Active sprints** page.
  - c. If you don't have any bugs on the board, go to the Backlog, create an issue of type bug, and add it to the sprint.
  - d. Click the three dots (...) above the board and select **Configure board**.
  - e. On the board configuration sidebar, click General.
  - f. Click **Edit Filter Query**.
    - **Note:** This takes you to the filter (called Filter for MEG board) in the issue navigator and shows all the issues that match the current JQL query.
  - q. If you don't see the JQL query, click Switch to JQL.
  - h. Update the JQL query by adding the highlighted part.
    project = MEG **AND issuetype != Bug** ORDER BY Rank ASC
  - i. Click **Search** and you now see all non-bug issues.
  - j. Click **Save** next to the name of the filter. You should see a confirmation that the filter was saved successfully.
  - k. Return to the Megastorm project's **Active sprints** page. You shouldn't see any bugs shown any more on the board, either here or in the Backlog.
  - l. Return to board settings and review how the filter query has changed.



- **Note:** There are three consequences to updating the board filter:
  - 1. Reports will only include issues that match the board filter.
  - 2. Epics and Versions will reflect only counts of issues that match the board filter.
  - 3. Issues created in the project which do not meet the board filter won't be visible on the board. Users will see a message that the issue was created but it is not currently visible.

### Exercise 3 Managing Releases and Versions

- in this exercise, you create a new project with sample data which will also create releases and versions. Only Jira administrators can create boards with sample data.
  - 1. Create a project that contains sample project data:
    - a. Navigate to your site URL and log in as the Jira administrator, **Dakota Jones**.
    - b. Navigate to the **Deep Space** project.
    - c. From the project sidebar, click the **board drop-down** and select + **Create board** (the + Create board link appears at the bottom of the board drop-down).

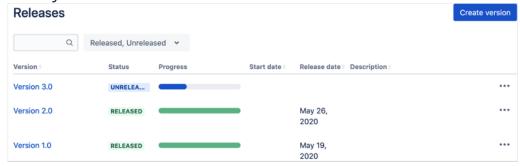


- d. Click Create a Scrum board with sample data.
  - i. Name the project **Caprica**.
  - ii. Change the project key to **CAP**.
  - iii. Leave the Project lead as Dakota Jones.
  - iv. Click **Create board**.
- e. After a short time you should see your new project's backlog which is full of issues.
- f. Set Max as the project administrator:
  - i. At the bottom of the project sidebar, click **Project settings** (or the cog icon).
  - ii. In the project settings sidebar, click **People**.
  - iii. On the People page, click Add people.
  - iv. Start typing **max** and select **Max Taylor**.
  - v. For role, select **Administrators**.
  - vi. Click Add 1 person.
- 2. View Releases and Versions:
  - a. Go to your site URL and log in as the project administrator, Max Taylor.
  - b. Go to the **Caprica** project.
  - c. In the project sidebar, click **Releases**.
    - **Note:** The Releases page shows the versions in this project and their status. Versions help you organize your work by giving you milestones to aim for. You can then assign the issues in your project to a specific version and build up the work you need to do to complete that version.
    - Question: How many unreleased versions does this project have?
    - Answer: This project has two unreleased versions versions 2.0 and 3.0.
  - d. Click the drop-down above the versions and select Released (leaving Unreleased also selected). Now you also see version 1.0 which has been released.

- Question: How many archived versions does this project have?
- **Answer**: Click the drop-down above versions again and select **Archived**. There are no archived versions in this project yet.
- e. Mouse over the different colors in the progress bar for version 2.0.
  - **Note:** Here you can quickly see how many issues are in what status in each version. Your team members can see most of the information on this page too.
- f. Click Version 2.0.
  - **Note:** This page is called the Release Hub and shows you all the issues and the status of each in this version. It gives you an up-to-date picture of release readiness.
- g. Click to open one of the issues.
  - **Note:** Note the Fix Versions fields. Once a version has been created for a project, this field becomes available for your issues.
- h. Click to edit the **Fix versions** field but don't change it.
  - **Note:** You can select from both unreleased and released versions to change the version this issue belongs to.
- i. Close the issue.
- 3. Release a version:

We need to release version 2.0 even though all issues are not done.

- a. Either click Back in your browser or return to the project's **Releases** page then click **Version 2.0**.
- b. Click **Release**:
  - i. Check Move unresolved issues to then select Version 3.0.
  - ii. For Release date, enter today's date.
  - iii. Click **Release**.
- c. Return to the **Releases** page and select Released from the drop-down if it's not already selected. The status of Version 2.0 is now **Released**.



- d. Click ... (three dots) for Version 2.0.
  - **Note:** To revert the release of a version, you can un-release the version here. You can also release unreleased versions here. Leave it as is.
- 4. Archive a version:

Version 1.0 has been out for a while now and we won't make any more changes to it so it's ready to be archived.

a. Click ... (three dots) next to Version 1.0 and select Archive.

- b. Select **Archived** from the drop-down and confirm status is now **ARCHIVED**.
- 5. Add a new version:
  - a. On the Releases page, click **Create version**.
    - i. Name: Version 4.0.
    - ii. Start Date: Enter today's date.
    - iii. Release date: A date in the future.
    - iv. Click **Save**.
  - b. Start to create a new issue in the Caprica project. Click the **Fix versions** drop-down and confirm Version 4.0 is now an option.
  - c. Cancel out of issue creation.
  - d. If you don't see all versions, select all statuses from the drop-down.



• **Note:** If you wanted to insert a new version 3.5 between 3.0 and 4.0, you could create the new version then simply drag the new version between them.

Congratulations on completing the lab!

### **Appendix**

### **Further Reading**

Reference	URL
Configure a company-managed board	https://support.atlassian.com/jira-software-cloud/docs/configure-a-company-managed-board/
Managing shared filters	https://support.atlassian.com/jira-cloud-administration/docs/manage-shared-filters/
Releases and versions	https://support.atlassian.com/jira-software-cloud/docs/enable-releases-and-versions/
View and manage a project's versions	https://support.atlassian.com/jira-work-management/docs/view-and-manage-a-projects-versions/

### **Best Practices**

Pitfall	Example Use Case	Best Practice
Certain columns may get overcrowded on the board. Reports are not accurately reflecting where issues are in the work process.	You use the default To Do, In Progress, and Done columns on a scrum board. But issues in the In Progress column could be in development or being tested. This column gets very full and the team lead can't see what issues are being worked on and what are being tested.	Configure the columns on your board to match how your team works.

Team members are frustrated because there are too many issues on the board.

Your team has 50 issues on the board. They can't quickly see the high priority issues or ones that matter to them. Create quick filters so your team can quickly see important issues. Update the board filter to permanently change what's shown on the board. Visually break up the board with swimlanes. Highlight certain issues with card colors.

### Lab 5 Boards & Projects

Scenario:

In these exercises, as the project administrator, you configure boards in kanban projects. You create multiple boards in one project. In another project you configure the board to show issues from multiple projects.

### Required for the lab

- 1. If you're using a University cloud site, you can use two projects that already exist on the site, Enterprise and Voyager. If you're not using the University cloud site, you'll need to create the two projects and change the owner of one of the board filters:
  - a. Log in to your site as the Jira administrator, Dakota Jones.
  - b. On the Projects page, click **Projects** in the top navigation bar and select **Create**
  - c. If you see the **Project templates** screen, follow the instructions in this step, then skip to **step e**.
    - i. Make sure **Software development** is selected in the sidebar.
    - ii. Select Kanban.
    - iii. Click **Use template**.
    - iv. Click Select a company-managed project.
    - v. Name the project **Enterprise**.
    - vi. Leave **Share settings with an existing project** unchecked.
    - vii. Click Create project (if you see Next instead, select that, and on the following screen, choose **Skip**.)
  - d. If you did not see the **Project templates** screen, follow these instructions:
    - i. Name the project Enterprise and select the Software Kanban template.
    - ii. Leave Share settings with an existing project unchecked.
    - iii. Click Create
  - e. At the bottom of the project sidebar, click **Project settings** (or the cog icon).
  - f. In the project settings sidebar, click **People**.
  - g. On the People page, click Add people.
    - i. Start typing **max** and select **Max Taylor**.
    - ii. For role, select **Administrators**.
  - h. Click Add.
  - i. Repeat these steps to create a project called **Voyager**.
  - j. Change the owner of one of the board filters:
    - i. Go to the Shared filters Jira administration page by using the . (period)
    - ii. In the **Filter for Voyager board** row, click ... (three dots) and change the owner to **Max Taylor**.

### Exercise 1 Creating Multiple Boards in a Project

- ① Your team is developing a new game app. When your developers are working on bugs they use kanban (a continuous flow). When they work on stories and tasks, they use sprints. You need a board for each of these. You also want a board where you can see all the issues in the project.
  - 1. View the project's default board:
    - a. Log in to your site as the project administrator, Max Taylor.
    - b. Go to the Enterprise project's Kanban board.
    - c. If you just created this project:
      - i. Create 2 bugs, 2 stories and 2 tasks. You can give them simple summaries such as Test bug 1 or use the ones in the screenshot below.
      - ii. Place one of each issue type in the Backlog and Selected for Development columns.
        - **Note:** This default kanban board shows all the issues in the project. We'll use this as our primary board.



- 2. Create a scrum board for stories:
  - a. At the top of the project sidebar, click the board drop-down and select **Create board**.
  - b. Click Create a Scrum board.
  - c. Ensure **Board from an existing project** is checked and click **Next**.
  - d. On the Name this board dialog:
    - i. Board name: Story board.
    - ii. Project: Enterprise.
    - iii. Location: Enterprise.
- 3. Click Create board.
  - **Note:** Now you see the backlog for the Story board. Note you see all the issues as you haven't created a sprint yet.
- 4. Create a sprint:
  - a. Click **Create sprint**.
  - b. Drag one issue of each type into the sprint.
  - c. Click **Start sprint**.
  - d. On the Start sprint dialog, accept the defaults and click **Start**.
    - Note: For this board we only want to see stories and tasks.

- e. On the Active sprints page, click the three dots (...) above the board and select **Configure board**.
  - i. On the General page, click Edit Filter Query.
  - ii. If you see the JQL query, click **Switch to basic**.
  - iii. Click **Type** drop-down and select the **Story** and **Task** issue types.
  - iv. Next to the filter name, click **Save**. You should see a message that the filter was successfully saved.
- f. Return to the **Enterprise** project.
- g. View the Story board's **Active sprints** and **Backlog** pages. On both, you should only see stories and tasks now.

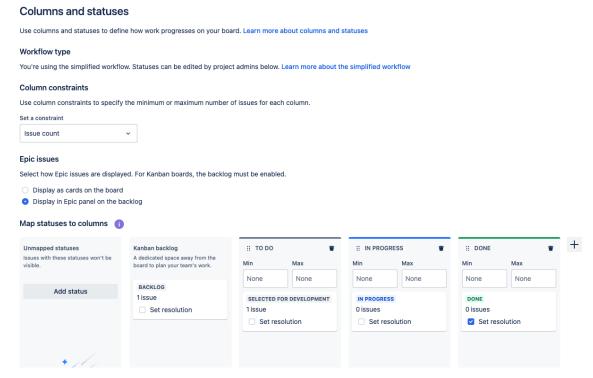


- 5. Create kanban board for bugs:
  - a. At the top of the project sidebar, click the board drop-down and select **Create** board.
  - b. Click **Create a Kanban board**.
  - c. Ensure **Board from an existing project** is checked and click **Next**.
  - d. On the Name this board dialog:
    - i. Board name: Bug board.
    - ii. Project: Enterprise.
    - iii. Location: Enterprise.
    - iv. Click Create board.
      - **Note:** Now you see all the issues in the project on a Kanban board. For this board we only want to see bugs.
  - e. Go to Configure board or Board settings:
    - i. On the General page, click Edit Filter Query.
    - ii. If you see the JQL guery, click Switch to basic.
    - iii. Click **Type** drop-down and select the **Bug** issue type.
    - iv. Click Save.
  - f. Return to the **Enterprise** project and view the Bug board (Kanban board). You should only see bugs now. There is no backlog so you see all bugs in the TO DO column.

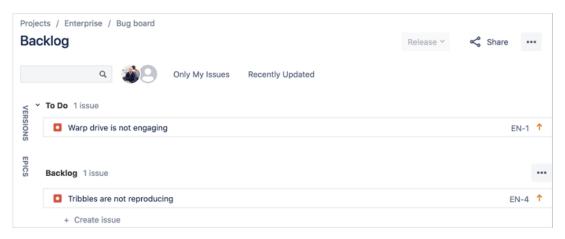


- 6. Enable kanban backlog:
  - a. On the Bug board, go to Configure board or Board settings.

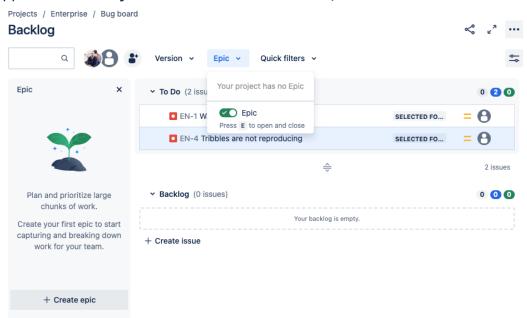
- b. Go to the **Columns** page.
- c. Optionally, if the project sidebar is expanded, click [ or the < at the top of the sidebar to collapse it and give you more space.
- d. Drag the **Backlog** status into the Kanban backlog to the left.
- e. Above the columns, click to enable the **Epics panel** so epics don't show as cards on the board.



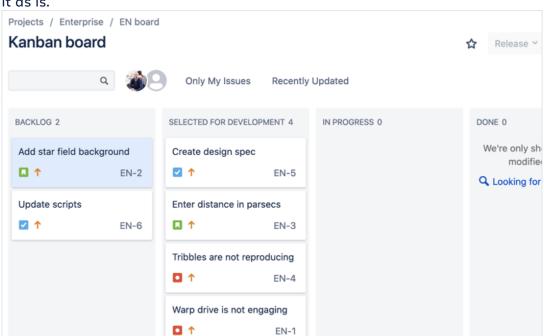
- f. Return to board. Now, you only see one bug as the other bug was moved to the new backlog.
- g. Either click **Take me there** in the popup dialog, or in the project sidebar, click the **Backlog**.
  - Note: Here, you see one of the bugs in Backlog and one in To Do. From here,
    the product owner can groom the backlog and move issues into the To Do
    column when they're ready for development. This is similar to a product owner
    or Scrum master moving issues from the Scrum backlog into the sprint
    backlog.



- h. Drag a bug from **Backlog** to **To Do**.
- i. Click the **Epic** dropdown and select Epic to open the Epic panel where you can create epics.
  - **Note:** The **EPICS** link used to be to the left of the backlog. The link text appeared vertically. This is now above the issues, as seen in the screenshot.



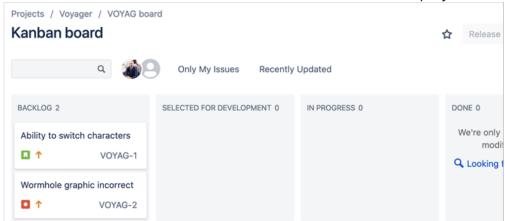
- j. Click **x** to hide the epic panel.
- k. In the project sidebar, click **Kanban board** to return to the Bug board, and now you will see both bugs in the To Do column.
- 7. View primary board:
  - a. Click the board drop-down and select **EN board**.
    - Note: By default, this default kanban project board shows all issues in all
      columns including both scrum and kanban backlogs. We could configure the
      board columns and move the Backlog status into the Kanban backlog. But the



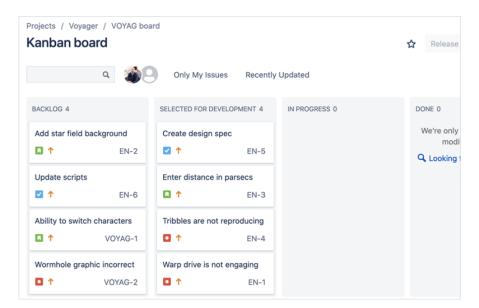
scrum master wants to be able to see everything on one board so we'll leave it as is.

### Exercise 2 Showing Multiple Projects on a Board

- (i) Max is managing two projects that have merged and he wants to see all the issues on one board.
  - 1. View multiple projects on a single board:
    - a. Log in to your site as the project administrator, Max Taylor.
      - **Note**: Your **Project settings** live in the project sidebar, to the left of your project. Since you are the project administrator you can can manage and configure a project there.
    - b. Go to Voyager project's Kanban board.
    - c. If you just created this project:
      - i. Create 2 new issues of any type. You can give them simple summaries such as Test bug 1 or use the ones in the screenshot below.
      - ii. Leave them both in the Backlog column.
        - Note: This default kanban board shows all the issues in the project.



- d. Go to **Configure board** or **Board settings**.
  - i. On the General page, click Edit Filter Query.
  - ii. If you see the JQL query, click Switch to basic.
  - iii. Click the project drop-down (first on left). Leave Voyager selected and also select **Enterprise**.
  - iv. Click Save.
    - **Note:** The Jira administrator who created the project, and who originally owned this board filter, has transferred ownership to Max.
- e. Return to the **Voyager** project where you now see issues from both projects on the board. Note the different project keys in the issue keys.



2. Configure swimlanes by project:

We set up swimlanes per project to make it easier to differentiate which issues come from which project.

- a. Go to **Configure board** or **Board settings**.
- b. On the board configuration sidebar, click **Swimlanes**.
- c. Click the **Select method** drop-down and select **Projects.**
- d. Return to the board, and now you can easily see which issues are from which project.

Congratulations on completing the lab!

## **Appendix**

# **Further Reading**

Reference	URL
What is a Jira Software board?	http://go.atlassian.com/jiraswboard
Create a board	http://go.atlassian.com/createjiraboard
Use your kanban backlog	http://go.atlassian.com/kanbanbacklog

#### **Best Practices**

Pitfall	Example Use Case	Best Practice
Too many issues to scroll through easily in one column.	Our team uses a kanban board but developers are complaining because there's too many issues in the To Do column to scroll through easily.	Managing your backlog in the first column of your Kanban board is easy to do – as long as there are only a few issues in your backlog. As your backlog grows, viewing and scrolling through these issues can become difficult. Enabling the kanban backlog gives you a separate backlog page where you can create and rank issues for your team.
Sprints can be unnecessary for certain issue types in your project.	Our team uses sprints to develop new features but when fixing bugs we simply use a continuous to do list.	If you use different agile practices for different issue types in the same project, create separate scrum and kanban boards and edit the

	board filters to show just those issue types.
--	---

### Lab 6 Managing Issues



Scenario:

In these exercises you configure issue types and components and use bulk change to change labels in issues and also to move issues. You perform some tasks as the Jira administrator and others as the project administrator.

#### Required for the lab

- If you completed the first lab in this course, you can use the existing Megastorm project. If you didn't create this project, you need to create a new one:
  - a. Log in to your site as the Jira administrator, **Dakota Jones**.
  - b. On the Projects page, click **Projects** in the top navigation bar and select **Create** project.
  - c. If you see the **Project templates** screen, follow the instructions in this step, then skip to **step d**.
    - i. Make sure **Software development** is selected in the sidebar.
    - ii. Select Scrum.
    - iii. Click **Use template**.
    - iv. Click Select a company-managed project.
    - v. Name the project Megastorm.
    - vi. Leave Share settings with an existing project unchecked.
    - vii. Click Create project (if you see Next instead, select that, and on the following screen, choose **Skip**.)
  - d. If you did not see the **Project templates** screen, follow these instructions:
    - i. Name the project **Megastorm** and select the Software **Scrum** template.
    - ii. Leave **Share settings with an existing project** unchecked.
    - iii. Click **Create**.
  - e. At the bottom of the project sidebar, click **Project settings** (or the cog icon).
  - f. In the project settings sidebar, click **People**.
  - g. On the People page, click Add people.
    - i. Start typing **max** and select **Max Taylor**.
    - ii. For role, select **Administrators**.
    - iii. Click Add.

### Exercise 1 Configuring Issue Types

- ① Your team wants to track feature requests separately from bugs so we'll create a new Feature Request issue type. Only the Jira administrator can create new issue types.
  - 1. Create a new issue type:
    - a. Go to your site URL and log in as the Jira administrator, **Dakota Jones**.
    - b. Go to the **Megastorm** project.
    - c. Click **Project settings** (or the cog icon) in the project sidebar and view the **Summary** page.
      - Question: What are the issue types for this project?
      - Answer: The issue types for this Software Scrum project are Bug, Epic, Story, Sub-task, and Task.
    - d. Click **MEG: Scrum Issue Type Scheme** which takes you to the Issue types page in the project.
    - e. Click Actions Edit issue types.
      - **Note:** This takes you to the Modify Issue Type Scheme page for the project in Jira administration.
    - f. Click + Add Issue Type:
      - i. Name: Feature Request.
      - ii. Description (optional): Features users would like to see.
      - iii. Type: Standard Issue Type.
      - iv. Click Add.
    - g. Drag the issue types to change the order so **Story** then **Bug** then **Feature Request** appear at the top of the list.
      - **Question:** What is the default issue type when creating an issue for this project?
      - **Answer**: The default issue type is Story. It's shown in the Default Issue Type field. You can choose another issue type here but we'll leave it as Story.
    - h. Click Save.
    - i. Confirm your new Feature Request issue type is now listed in the current issue type scheme for the Megastorm project.

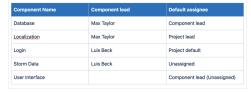


#### 2. Test issue types:

- a. Start to create a new issue for the **Megastorm** project.
- b. Confirm the default issue type is **Story**.
- c. Click the **Issue Type** drop-down and confirm **Bug** and **Feature Request** appear next in order.
- d. Select **Feature Request** for the Issue Type.
- e. Enter a Summary e.g. Test 4.
- f. Click Create.
  - **Note:** The Jira administrator can change the avatar (icon) for an issue type by editing it on the Issue Types Jira administration page.

### Exercise 2 Configuring Components

- (i) Max wants to start using components in his Megastorm project, so he can group the issues and have them automatically assigned to the appropriate developer.
  - 1. Create components:
    - a. Go to your site URL and log in as the project administrator, **Max Taylor**.
    - b. Go to the **Megastorm** project settings **Components** page.
    - c. If you see information about Compass, click on the **Compass components** button in the top right and select **Switch to Jira components**.
    - d. Click Create component:
      - i. Name: Database.
      - ii. Component Lead: Max Taylor.
      - iii. Default Assignee: Component Lead (Max Taylor).
      - iv. Click Save.
    - e. Click the ... (three dots) in the row for the new component. You can edit or delete the component here. Don't do that now.
    - f. Create four more components using the remaining entries in this table:



g. Your components should look like this:



- 2. Test auto-assignment of components:
  - a. Try to answer the following questions then create stories to test auto-assignment (use the table above for reference):
    - **Question:** If you created an issue and selected the **Database** component, who would the issue be assigned to?
    - Answer: It would be assigned to Max Taylor, the component lead.
    - **Question:** If you created an issue and selected the **Localization** component, who would the issue be assigned to?
    - **Answer**: Even though Max Taylor is the component lead, the project lead is set as the default assignee, so the issue is assigned to Luis Beck.
    - **Question:** If you created an issue and selected the **Login** component, who would the issue be assigned to?

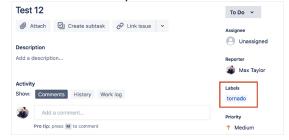
- **Answer**: Even though Luis Beck is the component lead, the project default is set as the default assignee. As the default assignee for the project is set to Unassigned, this issue is also set to Unassigned.
- Question: If you created an issue and selected the Storm Data component, who would the issue be assigned to?
- **Answer**: Even though Luis Beck is the component lead, the default assignee is set to Unassigned, so the new issue is also set to Unassigned.
- Question: If you created an issue and selected the User Interface component, who would the issue be assigned to?
- **Answer**: Even though the Default Assignee is set to the component lead, because there is no component lead defined for this component, the new issue is set to Unassigned.
- **Question:** If you created an issue and selected the **Database** component and set the Assignee to Luis Beck, would the issue stay assigned to Luis or change to Max Taylor who's the default assignee (component lead)?
- **Answer**: Note that when you create the issue and view the Assignee field, it initially shows Automatic. By selecting a specific user, you override the autoassignment. It would stay assigned to Luis Beck.
- **Question:** If you created an issue without selecting a component or an assignee then later went back and updated the component to **Database**, would the assignee change to Max (component lead)?
- **Answer**: The Assignee would stay as Unassigned.

### Exercise 3 Using Bulk Change to Correct Labels

- ① Users are using incorrect label values when creating or updating issues. In this exercise you'll correct the incorrect labels using bulk change.
  - 1. Create test issues:
    - a. Create an issue in the **Megastorm** project:
      - i. Use **Story** issue type and any summary.
      - ii. In the Labels field enter tornado and select tornado (New Label).
      - iii. Click **Create**.
    - b. Repeat these steps to create two more issues but use the misspelled labels **tornad** in one issue and **toranado** in the other issue.
- 2. Find issue with incorrect labels:
  - a. Place your cursor in the search field in the main menu and enter return to go to the Search page.
  - b. If you're on basic search, click **Switch to JQL** to go to advanced search.
  - c. Enter the JQL:

#### project = MEG AND labels in (tornad, toranado)

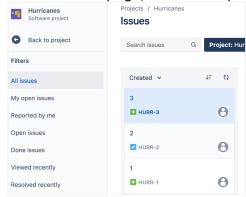
- Note: Don't include the correct label, tornado.
- d. Click **Search**. You should see the two issues with incorrect labels.
- 3. Bulk change labels:
  - a. Click the ... (three dots) on the top right of the search page and select **Bulk change** all 2 issue(s).
  - b. Click the checkbox on the left above the issues to select all the issues and click **Next**.
  - c. Select Edit Issues and click Next.
  - d. Scroll down the Operation Details page and select Change Labels:
    - i. From the drop-down select Replace all with.
    - ii. Start typing tor and select the correct label, tornado.
  - e. Scroll down and click Next.
  - f. On the Confirmation page, confirm you see your two issues and click either of the **Confirm** buttons to confirm.
  - g. When the bulk operation is 100% complete, click **Acknowledge**.
  - h. Search for and open both the issues to confirm the label was changed to tornado.



### Exercise 4 Using Bulk Change to Move Issues from One Board to Another

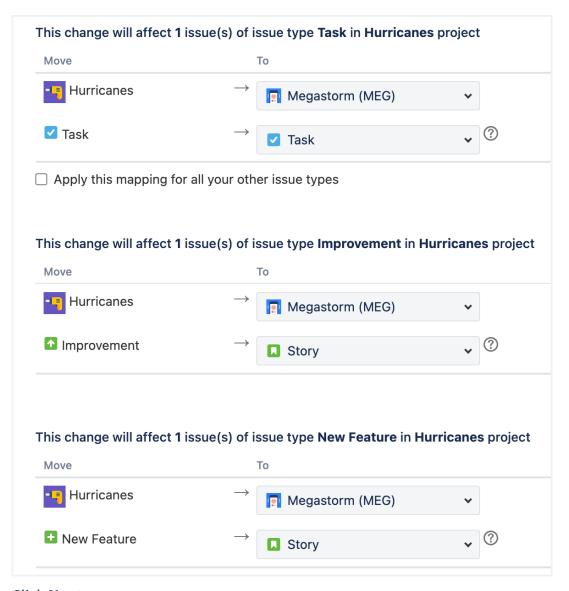
- ① Two of your projects, Hurricanes, and Megastorm, have merged and you've decided to move the issues from one project to another. However, the projects use different project templates and have different issue types. In this exercise, we'll move the issues without losing any of the data.
  - 1. Set up the project to move issues from:
    - a. Log in to your site as the Jira administrator, **Dakota Jones**.
- 2. Create a company-managed project:
  - a. From the Projects drop-down menu, select Create project.
  - b. If you see the **Project templates** screen, **follow the instructions in this step**, then skip to **step 5**.
  - c. Make sure **Software development** is selected in the sidebar.
  - d. Select **Bug tracking**.
  - e. Click **Use template**.
  - f. Name the project Hurricanes and the project key at HURR.
  - g. Click Create project.
- 3. If you did not see the **Project templates** screen, follow these instructions:
  - a. Click **Create**.
  - b. Select Company-managed project.
  - c. Name the project Hurricanes and the project key, HURR.
  - d. Click the drop-down at the top of the page. Here you can filter project templates based on product. If it's not already selected, select **Software**.
  - e. Here you see all the project templates for Jira Software. Select the Software **Bug tracking** template.
  - f. Leave **Share settings with an existing project** unchecked.
  - g. Click **Create**.
- 4. If you are taken to the Open issues page, click **Back to project**.
- 5. At the bottom of the project sidebar, click **Project settings** (or the cog icon).
- 6. In the project settings sidebar, click **People**.
- 7. On the People page, click Add people.
  - a. Start typing max and select Max Taylor.
  - b. For role, select Administrators.
  - c. Click **Add 1 person**.
- 8. View project issue types:
  - a. In the **Hurricanes** project, go to the **Issue** and then **Types** project settings page.
  - b. Note the issue types for this Software Bug tracking project **Bug, Epic, Improvement, New Feature, Task,** and **Sub-task**.
  - c. Go to the **Megastorm** project and view the **Issue** and then **Types** project settings page.
  - d. Note the issue types for this Software Scrum project **Story**, **Bug**, **Epic**, **Feature Request** (you'll see this if you did an earlier exercise), **Task**, and **Sub-task**.

- Question: What issue types are different between the two projects?
- Answer: The Bug tracking project has the Improvement and New Feature issue types which are not in the Scrum project. The Scrum project has the Story and Feature Request issue types which are not in the Bug tracking project.
- 9. Create issues to move:
  - a. Go to your site URL and log in as the project administrator, Max Taylor.
  - b. Go to the **Hurricanes** project.
  - c. Create three issues in the **Hurricanes** project: A Task, A New Feature, and an Improvement.
  - d. On the Issues page, refresh the page or click All issues.



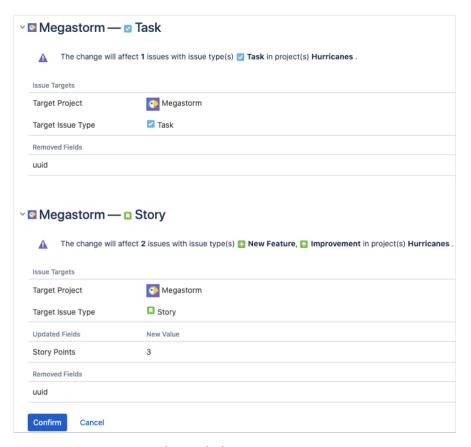
#### 10. Bulk move the issues:

- a. Click **Go to advanced search** to go to the Search page.
- b. Click the ... (three dots) on the top right of the search page and select **Bulk change** all 3 issue(s).
- c. Click the checkbox on the left above the issues to select all the issues and click **Next**.
- d. Select Move Issues and click Next.
- e. For each issue:
  - i. Select the **Megastorm** project to move to.
  - ii. For the **Task** issue type, select **Task**.
  - iii. For the Improvement and New Feature issue types, select Story

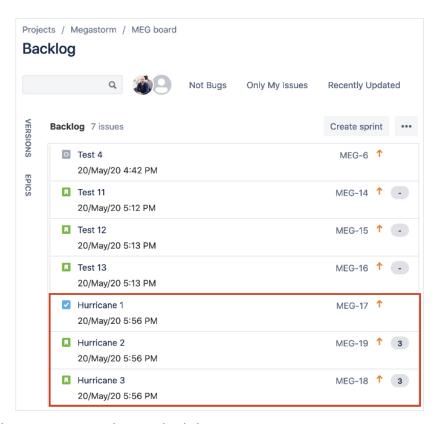


#### iv. Click **Next**.

- f. On the Step 3 of 4: Update fields for 'Task' issues in the 'Megastorm' project page, click **Next**.
- g. On the Step 3 of 4: Update fields for 'Story' issues in the 'Megastorm' project page:
  - **Note:** Here you have the option to enter a value for Story Points as you're migrating two issues to the Story issue type.
    - Enter a Story Points field value of 3 and click Next.
      - On the Confirm changes page, confirm that 1 Task, 1 New Feature, and 1 Improvement in the Hurricanes project will be affected and click Confirm.



- h. When the bulk operation is 100% complete, click **Done**.
- 11. Confirm the issues were moved:
  - a. Return to the **Hurricanes** project and confirm the issues are no longer there.
  - b. Go to the **Megastorm** project and view the **Backlog**.
  - c. Confirm you see your three hurricane issues and that the two stories have story points set to 3.



Congratulations on completing the lab!

# Appendix

## **Further Reading**

Reference	URL
Adding, editing, and deleting an issue type	https://support.atlassian.com/jira-cloud-administration/docs/add-edit-and-delete-an-issue-type/
Organize work with components	https://support.atlassian.com/jira-software-cloud/docs/organize-work-with-components/
Edit multiple issues at the same time	https://support.atlassian.com/jira-service-management-cloud/docs/edit-multiple-issues-at-the-same-time/

#### **Best Practices**

Pitfall	Example Use Case	Best Practice
Not able to easily differentiate different types of issues when you use the same issue type.	Our team is tracking both bugs and feature requests using the Bug issue type. But the team lead can't see how many of each we're actually working on.	If the default issue types don't meet your needs, create new issue types to reflect the work items you use.

### Lab 7 Automation

③ Scenario: In this exercise, as the project administrator, you create an automation rule to automate work in your project.

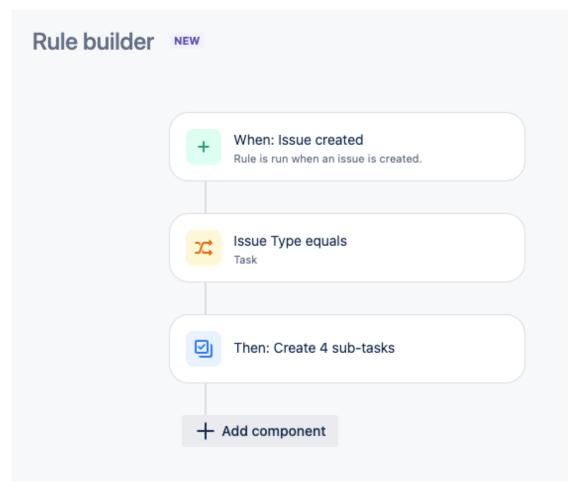
### Exercise 1 Creating an Automation Rule

- (i) Your team is now using issues of type Task to create blog posts. When they create blog posts they need to perform four subtasks. To make it easier, they want to have the four subtasks created automatically when they create these types of issues. You can do this by creating an automation rule.
  - 1. Create a project:
    - a. Log in to your site as the Jira administrator, **Dakota Jones**.
    - b. Click Projects in the top navigation bar and select Create project.
    - c. If you see the **Project templates** screen, **follow the instructions in this step**, then skip to **step e**.
      - i. Make sure **Software development** is selected in the sidebar.
      - ii. Select Kanban.
      - iii. Click **Use template**.
      - iv. Click Select a company-managed project.
      - v. Name the project **Tardis**.
      - vi. Leave **Share settings with an existing project** unchecked.
      - vii. Click Next.
      - viii. If you see the 'Connect your work' page, click Continue.
    - d. If you did not see the **Project templates** screen, follow these instructions:
      - i. Name the project **Tardis** and select the Software **Kanban** template.
      - ii. Leave Share settings with an existing project unchecked.
      - iii. Click **Create project**.
    - e. At the bottom of the project sidebar, click **Project settings** (or the cog icon).
    - f. In the project settings sidebar, click **People**.
    - g. On the People page, click Add people.
      - i. Start typing max and select Max Taylor.
      - ii. For role, select **Administrators**.
    - h. Click **Add 1 person**.
- 2. Explore automation rules:
  - a. Go to your site URL and log in as the project administrator, Max Taylor.
  - b. Go to the **Tardis** project.
  - c. At the bottom of the project sidebar, click **Project settings** (or the cog icon).
  - d. In the project settings sidebar, click Automation.
  - e. Click the Rules tab (if required) and then click Create rule.
  - f. You can view the short Let's create your first rule tour if you would like by clicking **Next**, or you can click **Skip tour**.
    - i. If you have completed the tour, click **Get started**.
  - g. Scroll down to view the triggers that are available to you.
    - **Note:** Triggers start the execution of a rule. They can be based on an event that occurs, scheduled to run (see the Scheduled trigger), or manually triggered (see the Manual trigger).

- 3. Create a trigger that executes when an issue is created:
  - a. Under Add a trigger, click the **Issue created** trigger.
  - b. Click Next.
- 4. Create a condition that only selects issues of type Task:
  - a. Under Add component, click IF: Add a condition.
  - b. Select Issue fields condition.
    - i. Field: **Issue Type**.
    - ii. Condition: **equals**.
    - iii. Value: Task.
    - iv. Click **Next**.
- 5. Create an action that automatically creates four subtasks:
  - a. Under Add component, select **THEN: Add an action**.
  - b. Select Create sub-tasks.
    - i. For Summary, enter **Write blog post**.
    - ii. Click **Add another sub-task**. (If you accidentally click Save to save the action, click on the action to edit it.)
    - iii. Repeat this process to create three more subtasks with summaries:
      - 1. Create graphics.
      - 2. Approve blog post.
      - 3. Submit blog post.
    - iv. Click Next.
- 6. Enable your automation rule:

When: Issue created, Issue Type equals Task, Then: Create 4 sub-tasks.

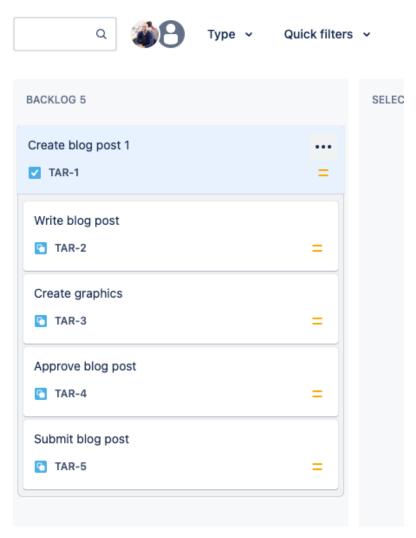
a. Verify that under Automation, your automation rule now lists:



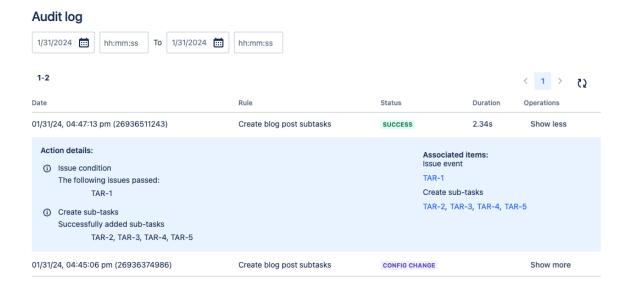
- b. Click **Turn on rule** and name your rule **Create blog post subtasks,** then click **Turn on rule**. Your automation rule is now turned on.
- c. In the top right, click **Rule details** and explore the rule's details.
  - Question: Can you use this automation rule in another project?
  - **Answer**: The scope for this rule is this project so you can't use it in another project. However, the Jira administrator can set up global rules on the Automation rules Jira administration page.
- 7. Test your automation rule:
  - a. Click **Back to project**.
  - b. Create an issue that triggers the rule:
    - i. Project: Tardis.
    - ii. Issue Type: **Task**.
    - iii. Summary: Create blog post 1.
    - iv. Click **Create**.
  - c. View your Kanban board. You should see that the automation rule created four subtasks for your issue. If you don't see the subtasks, refresh your browser window.

Projects / Tardis / TAR board

#### TAR board



- 8. View the automation audit log:
  - a. Return to the Project settings **Automation** page.
  - b. Click on your **Create blog post subtasks** rule.
  - c. On the right click **Audit log**.
  - d. You should see two entries one for the creation of the rule and one for the successful execution of the rule. Click **Show more** next to this entry.



#### 9. Disable your automation rule:

Your team is no longer creating blog posts, so they want to disable the automation rule. But they want to keep it in case they start creating blogs again.

- a. At the top of the page click Return to rules.
- b. To disable your rule, click the green checkmark under Enabled for your rule.

Congratulations on completing the lab!

## **Appendix**

## **Further Reading**

Reference	URL
Automation: Basics	http://go.atlassian.com/jiraautomationbasics
Jira automation	www.atlassian.com/software/jira/features/automation
Smart values in automation rules	https://support.atlassian.com/cloud- automation/docs/what-are-smart-values/
Rule actor in automation rules	https://support.atlassian.com/cloud- automation/docs/what-is-a-rule-actor/
Getting started with automation in Jira webinar	http://go.atlassian.com/ jiraautomationwebinar

### **Best Practices**

Pitfall	Example Use Case	Best Practice
Automation rule had unexpected consequences.	You created a rule to create subtasks but you forgot to create a condition to execute it for just the Task issue type. Now it's creating subtasks for every issue type in the project.	Ensure you thoroughly test your automation rules when you create them.

# Lab 8 Reports & Dashboards

(i) Scenario:

In these exercises, as the project administrator, you create a new dashboard and add gadgets to it. You also run a number of reports.

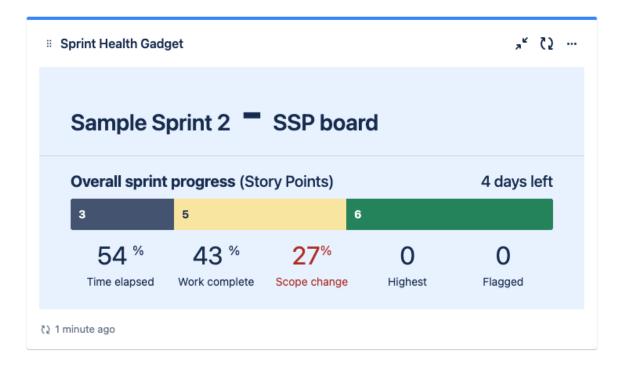
#### Exercise 1 Creating a Dashboard

- i First, you create a new project with sample data which creates a couple of sprints. Only Jira administrators can create projects with sample data.
  - 1. Create a new project with sample data:
    - a. As the Jira administrator, **Dakota Jones**, navigate to any company-managed software type project (such as **Enterprise**).
    - b. From the project sidebar, click the board drop-down and select **Create board**.
    - c. Click Create a Scrum board with sample data.
      - i. Use the default project name of **Sample Scrum Project** or change it to any name you wish. These instructions will use the default name.
      - ii. Click Create board.
        - Note: After a short time you see your new project's backlog which is full
          of issues.
    - d. Set Max as the project administrator:
      - i. At the bottom of the project sidebar, click **Project settings** (or the cog icon).
      - ii. In the project settings sidebar, click **People**.
      - iii. On the People page, click Add people.
      - iv. Start typing max and select Max Taylor.
      - v. For role, select **Administrators**.
      - vi. Click Add 1 person.
- 2. Create a dashboard:

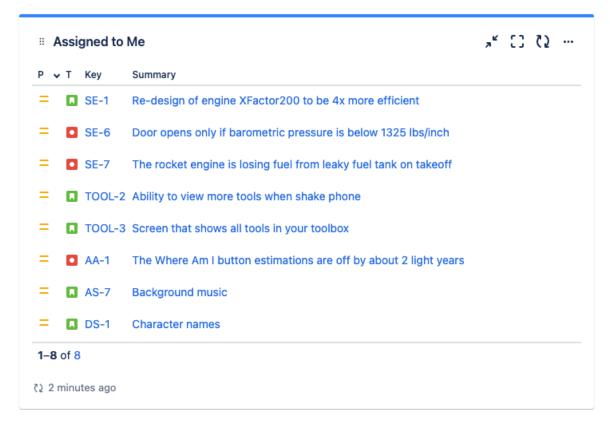
Max wants to create a dashboard that will be useful to members of the team. He wants them to be able to get a quick status on the progress of the current sprint, what issues are assigned to them, and view the issues in the sprint.

- a. Go to your site URL and log in as the project administrator, Max Taylor.
- b. In the main menu, click **Dashboards** and select **Create dashboard**.
  - i. Name: Development dashboard.
  - ii. Click the Viewers drop-down and select Project.
  - iii. Click the Project field then search for and select Sample Scrum Project.
  - iv. Leave the roles at All roles.
  - v. Click Add.
  - vi. Leave Editors as Private/Only you.
  - vii. Click Save.
- 3. Add the Sprint Health gadget to the dashboard:
  - a. If necessary, in the left column click **add a new gadget** to open the **Add a Gadget** panel on the right.
    - i. Ensure the **All** filter is selected under the search box.
    - ii. Scroll down and view all the gadgets that are available to you.
    - iii. Find the **Sprint Health Gadget** and click **Add**.
    - iv. Close the Add a gadget dialog.
  - b. Configure the Sprint Health Gadget:

- i. Board: SSP board.
- ii. Sprint: **Sample Sprint 2**.
- iii. Auto refresh: Check Update every 15 minutes.
  - **Note:** Auto refresh is useful if you view a dashboard often and you need up to date statistics. But be careful about setting too many gadgets to auto refresh to avoid strain on your Jira site.
- iv. Click **Save**.
- c. Now you see the gadget displayed on your dashboard.



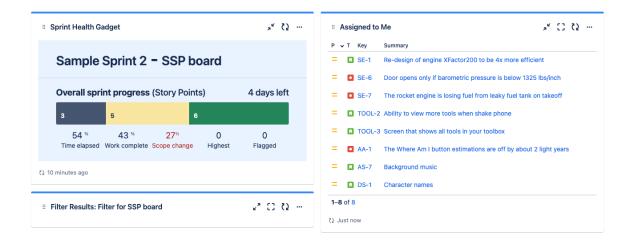
- 4. Add Assigned to Me gadget.
  - a. In the right column, click add a new gadget to reopen the Add a Gadget panel.
    - i. Find the **Assigned to Me** gadget and click **Add**.
    - ii. Close the Add a Gadget panel.
  - b. Configure the Assigned to Me gadget:
    - i. Drag the **Priority** field to the top of the Columns list to display.
    - ii. Click the drop-down and select the **Fix versions** field.
    - iii. Click **Save**.
  - c. In the Assigned to Me gadget, click ... (three dots) and select **Configure**.
    - i. You want to remove the Fix versions field so click the trash icon next to it.
    - ii. Click **Save**.
  - d. Now, you see the gadget displayed on your dashboard.



#### 5. Edit the dashboard layout:

You can choose different column layouts to match the size of your gadgets and how many you want to place on the dashboard.

- a. On your dashboard, click **Change layout**.
- b. Experiment with choosing different column layouts to see how the display changes.
- c. Return to the default equal sized two column layout.
- 6. Add the Filter Results gadget:
  - a. At the top of the dashboard, click Add gadget:
  - b. Add the **Filter Results** gadget and close the dialog.
  - c. Configure the Filter Results gadget:
    - i. Saved Filter: Filter for SSP board.
    - ii. Click Save.
  - d. Change the size of the gadget:
    - i. To see a wider view of the filter results, click the **Maximize** icon (square) in the gadget header.
    - ii. To restore the gadget to its original size, click the **Single column** icon (plus sign) in the gadget header.
    - iii. To hide the results, click the **Minimize** icon (two diagonal arrows) in the gadget header.
  - e. Grab the Filter Results gadget and drag it around the dashboard to see where you can place it. Place it below the Sprint Health Gadget.



### **Exercise 2 Running Reports**

- 1. Run the Burndown Chart report:
  - a. Go to the Sample Scrum Project.
  - b. View the **Active sprints** page.
    - Note: On the Active sprints page you see Sample Sprint 2 which is still in progress. Sample Sprint 1 has already been completed.
  - c. In the project sidebar, click Reports.
  - d. Click **Burndown Chart**.
    - **Note:** The Burndown Chart tracks the completion of work throughout the sprint.
    - Question: How many story points are still to be completed in this sprint?
    - **Answer**: Look at the last position of the red line (remaining values) and see what the value it matches in the vertical y-axis.
  - e. Click the **Story Points** drop-down and select **Issue Count**.
    - Note: Now you can see the work done and remaining in number of issues.
  - f. Uncheck **Show Non-Working Days** in the top right of the chart.
    - Note: Now you can see the work done and remaining in number of issues.
    - **Question:** How many working days are left in this sprint? Is the team on track to meet the sprint goal?
    - **Answer:** Look at the last position of the red line (remaining values) and see what day it matches in the horizontal x-axis then count the days until the end of the sprint. So far the team is on track as the remaining values is close to the guideline.
  - g. Click the sprint drop-down and select Sample Sprint 1.
    - Note: Here you can see how the team performed in sprint 1.
  - h. Scroll down and explore the details in the chart at the bottom.
- 2. Run the Velocity Chart report:
  - a. In the Reports sidebar, click **Velocity Chart**.
    - **Note:** The Velocity Chart tracks the amount of work completed from sprint to sprint. This lets you predict the amount of work the team can get done in future sprints. Only one sprint has been completed so you only see the data for that sprint.
  - b. Return to the **Active sprints** page.
  - c. Move all the issues and subtasks to the **DONE** column. If prompted, click **Update** to update the parent issue.
  - d. Click Complete sprint, then Complete sprint again.
    - **Note:** You may be taken to the Sprint Report, which includes a burndown chart and a list of issues completed in the sprint. It's useful for sprint retrospective meetings.
  - e. In the Reports sidebar, click **Velocity Chart**. Now you see both sprints' data.
    - **Question:** What's the estimation statistic used for estimating stories by the team?

- **Answer**: You can see from the y-axis that the team is using story points to estimate stories.
- Question: In which sprint did the team complete less than they committed to?
- **Answer**: In Sample Sprint 1 the team completed slightly less than they committed to at the beginning of the sprint.
- **Question:** Based on these two sprints, how many story points do you think they should commit to in sprint 3?
- **Answer**: To calculate the velocity you average the total completed estimates. For example if they completed 16 story points in the first sprint and 14 story points in the second sprint, a good estimate for the next sprint is 15 story points.
- f. If you scroll down to the bottom, you can click each sprint to go to the sprint report.
- 3. Run the Control Chart:
  - a. In the Reports sidebar, click **Control Chart**.
  - b. Scroll down and, under Time scale, change the Timeframe to **Past Week**.
    - **Note:** The control chart shows the cycle time for issues in your sprint. The default display shows the time from In Progress to Done.
    - **Question**: Does the rolling average (blue line) indicate the team is increasing their efficiency and throughput?
    - **Answer**: No, the rolling average in this control chart shows that issues are taking longer as time progresses so efficiency is decreasing and there is less throughput.
    - Question: Which issues are worth investigating on this chart?
    - **Answer**: You want to look at the outliers on the chart, that is the issues outside the standard deviation (blue shaded area).
  - c. Click one of the hollow dots on the graph to view a single issue's details.
  - d. Click one of the filled dots on the graph to view data on a cluster of issues.
  - e. Drag the cursor along a portion of the chart to zoom in on that area of the chart.
  - f. Under Refine report, explore the **Columns** and **Quick Filters** and how you can change the data reported on the chart.

Congratulations on completing the lab!

## **Appendix**

## **Further Reading**

Reference	URL
Configuring custom dashboards	https://support.atlassian.com/jira-cloud-administration/docs/configure-custom-dashboards/
View and understand the burndown chart	https://support.atlassian.com/jira-software- cloud/docs/view-and-understand-the- burndown-chart/
View and understand the velocity chart	https://support.atlassian.com/jira-software- cloud/docs/view-and-understand-the- velocity-chart/
View and understand the control chart	https://support.atlassian.com/jira-software- cloud/docs/view-and-understand-the- control-chart/
View and understand the version report	https://support.atlassian.com/jira-software- cloud/docs/view-and-understand-the- version-report/
View and understand the sprint report	https://support.atlassian.com/jira-software-cloud/docs/view-and-understand-the-sprint-report/

#### **Best Practices**

Pitfall	Example Use Case	Best Practice
Having to go to many different places to find the data you're interested in.	Each day when I start work, I want to see which issues are assigned to me, the overall status of the current sprint,	Create dashboards so you have one place to give you an overview of the data

	and how the team is performing overall. I have to go to different pages in the project to find out this information.	you're interested in in one place.
Not being able to estimate with any certainty how much your team can accomplish in a sprint.	Your team has run a number of sprints now but you always misjudge how much the team can accomplish in each sprint.	Run reports to see how your team is performing in sprints and to optimize future sprints.

#### Lab 9 Other Jira Features

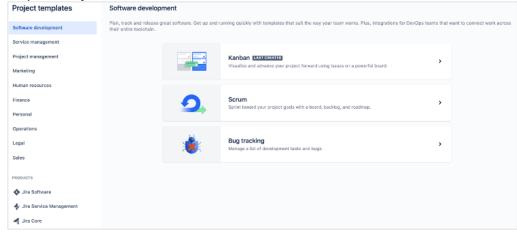
in these exercises you create a new project that shares its project configuration with another project and you run parallel sprints. You perform some tasks as the Jira administrator and others as the project administrator.

#### Required for the lab

- For this lab you need an existing project with the permission scheme changed. If you
  completed the earlier labs in this course, you can use the existing **Megastorm** project. If
  you didn't, do the following:
  - a. Log in to your site as the Jira administrator, Dakota Jones.
  - b. On the Projects page, click **Projects** in the top navigation bar and select **Create project**.
  - c. If you see the **Project templates** screen, **follow the instructions in this step**, then skip to **step e**.
    - i. Make sure **Software development** is selected in the sidebar.
    - ii. Select Scrum.
    - iii. Click **Use template**.
    - iv. Click Select a company-managed project.
    - v. Name the project **Megastorm**.
    - vi. Leave **Share settings with an existing project** unchecked.
    - vii. Click **Create project** (if you see **Next** instead, select that, and on the following screen, choose **Skip**.)
  - d. If you did not see the **Project templates** screen, follow these instructions:
    - i. Name the project **Megastorm** and select the Software **Scrum** template.
    - ii. Leave **Share settings with an existing project** unchecked.
    - iii. Click **Create**.
  - e. Change the permission scheme.
    - i. In the project settings sidebar, click **Permissions**.
    - ii. Click Actions and select Use a different scheme.
    - iii. Select **Development Permission Scheme** and click **Associate**.

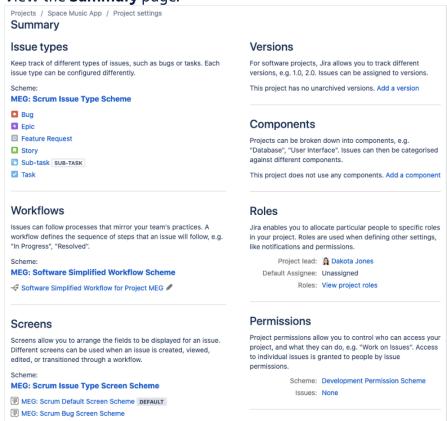
#### Exercise 1 Creating a Project with Shared Settings

- ① You want a new project that's set up the same way as an existing project, Megastorm. In that project the permission scheme was changed. Rather than making the changes yourself, it's a good practice to share schemes when projects have the same requirements.
  - 1. View an existing scrum project:
    - a. Go to your site URL and log in as the Jira administrator, **Dakota Jones**.
    - b. Go to **Megastorm** project.
    - c. In the project sidebar, click **Project settings** (or the cog icon).
    - d. In the project settings sidebar, click **Summary** and view the names of the schemes used in this project.
      - **Note:** When a project is created they get their own set of schemes. Note the project key MEG at the beginning of the name of many of the schemes. For example MEG: Scrum Issue Type Scheme. By default, all new Software projects use the Default software scheme. However the permission scheme for this project was changed to the Development Permission Scheme.
- 2. Create a new project:
  - a. Click Projects in the top navigation bar and select Create project.
  - b. If you see the **Project templates** screen, **follow the instructions in this step**, then skip to **step 3**.



- i. Make sure **Software development** is selected in the sidebar.
- ii. Select Scrum.
- iii. Click **Use template**.
- iv. Click Select a company-managed project.
- v. Name the project **Space Music App**.
- vi. Click **Share settings with an existing project**.
- vii. Select the **Megastorm (MEG)** project.

- viii. Click Next.
- ix. If you see the Connect your work page then click Continue.
- c. If you did not see the **Project templates** screen, follow these instructions:
  - i. Name the project **Space Music App** and select the Software **Scrum** template.
  - ii. Click Share settings with an existing project.
  - iii. Select the Megastorm (MEG) project.
  - iv. Click Create.
- 3. View project configuration:
  - Go to the Space Music App's project settings.
  - b. View the **Summary** page.



- Question: Does this project use the same schemes as the Megastorm project?
- **Answer**: Yes, both the Space Music App and the Megastorm projects share the same schemes.
- **Question:** Which projects would be affected if you updated one of these schemes?
- **Answer**: Since you created the Music App project by sharing the Megastorm project's configuration they share the same schemes. An update to any one of these schemes will affect both projects.
- **Note:** the board from the original project is not copied over. Team members would need to create a new one (which will create a Backlog page).

### **Exercise 2 Running Parallel Sprints**

i You have two teams working from the same backlog. By enabling parallel sprints, each team can work on their own active sprint simultaneously.

- 1. Enable parallel sprints:
  - a. As the Jira administrator, **Dakota Jones**, click the cog icon (Settings) in the main menu and select **Products**.
  - b. In the Products sidebar, click Jira Software configuration.
  - c. If it hasn't already been enabled, check Parallel Sprints.
    - Note: By default parallel sprints are not enabled in Jira Cloud.



- 2. Run a parallel sprint:
  - a. Go to your site URL and log in as the project administrator, Max Taylor.
  - b. Go to the **Megastorm** project.
  - c. In the project sidebar, click **Backlog**. Note that sprint 1 is in progress.
  - d. Above the issues in the backlog, but below the open sprint, click Create sprint.
  - e. Drag some of the issues from the backlog into MEG Sprint 2.
  - f. Click **Start sprint**.
    - i. Accept the defaults and click **Start**.
  - g. Back on the **Active sprints** page, you now see all the issues in both active sprints.
  - h. Click the **Sprint** dropdown above the board and select **MEG Sprint 2**. Now you only see the issues in this one sprint.

Congratulations on completing the lab!

# Appendix

## **Further Reading**

Reference	URL
Create a new project	http://go.atlassian.com/createprojectcloud
Use parallel sprints	http://go.atlassian.com/ parallelsprintsjiracloud

#### **Best Practices**

Pitfall	Example Use Case	Best Practice
The Jira administrator spends a lot of time updating schemes for new projects.	All your development scrum projects have the same requirements in terms of permissions, issues types, etc. which are slightly different from the default settings. The Jira administrator needs to edit each project's schemes as they create new development scrum projects.	Share project settings between projects that have the same requirements and are expected to stay the same in the future.

### Lab 10 Creating & Configuring Team-Managed Projects



#### Scenario

You now need to create a team-managed project for a Teams In Space development project team. This is a small team that's starting a new software project. Here you log in as a development team member and perform the following tasks:

- Create a team-managed (formerly next-gen) Software project
- Control access to a project
- Add issue types
- Add fields for issue types
- Configure the board
- Customize project features

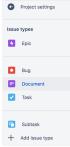
#### Exercise 1 Creating a Team-managed Project & Controlling Access

- 1. Create a team-managed Software project:
  - a. Log in to your cloud site as the development team member, Max Taylor.
  - b. If you're not already there, navigate to the Projects page by clicking **Projects** in the top navigation bar, then selecting **View all projects**.
  - c. On the Projects page, click **Create project** in the top right of the page.
  - d. If you see the **Project templates** screen, **follow the instructions in this step**, then skip to **step 2**.
    - i. Make sure **Software development** is selected in the sidebar.
    - ii. Select Kanban.
    - iii. Click **Use template**.
    - iv. Click Select a team-managed project.
    - v. Name the project Galaxy App and leave the key as GA.
    - vi. Make sure **Access** is set to **Open**.
    - vii. Click Click **Create project** (if you see **Next** instead, select that, and on the following screen, choose **Continue**.)
  - e. If you did not see the **Project templates** screen, follow these instructions:
    - i. Select **Try a next-gen project**.
    - ii. Name: Galaxy App
    - iii. Project access: Open
    - iv. Note the key, **GA**. Leave as is.
    - v. Click **Change template**.
    - vi. Select the Kanban template.
    - vii. Click Create.
- 2. Control who can work on the project:
  - a. Click **Project settings**, then **Access** in the sidebar.
    - Note: Max has the role of Administrator as he created the project. As the
      project has open access, you see a message saying that all registered users of
      the site have the member role. You don't need to specifically add members to
      this project. You also don't need to add viewers as all users already have the
      role of member.
- 3. Max's team decided that only users of Jira Software should be able to create and edit issues in their project.
  - a. Click **Change project access**.
  - b. Select **Limited**.
  - c. Click **Change**.
  - d. Click **Save and make limited**, to confirm the new settings.
    - Now anyone with access to the site has the viewer role. To add users who can add and edit issues, we need to add the users or group and assign them the role of member.
  - e. Click Add people.
    - i. Start typing **jira** and select the **jira-software-users** group.

- ii. Select the role, **Member**.
- iii. Click Add 1 person.
  - **Question:** What can a licensed user who's not a member of the jirasoftware-users group do in this project?
  - **Answer**: A user who's not in the jira-software-users group can search for and view issues in this project.

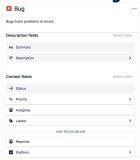
### Exercise 2 Adding & Configuring Issue Types

- 1. Add issue types:
  - a. Navigate to the Galaxy App project settings Issue types page.
    - Question: What are the default issue types for this Kanban project?
    - **Answer:** This Kanban project has the Epic, Task, and Subtask issue types by default. They are listed in the sidebar.
- 2. The team also needs to track bugs. Click **Add issue type** in the sidebar.
  - a. Bug is one of the suggested issue types and is already selected. Click Add.
    - **Note:** One of the team's responsibilities is to update the documentation when they make a change to the app. We'll create a new issue type for that.
- 3. Click Add issue type.
  - a. Click Create issue type.
  - b. Name: **Document**
  - c. Optional description: An update to the documentation.
  - d. Click **Change icon** then ... and select the purple one with lines (or another one if you wish).
  - e. Click **Select**.
  - f. Click Create.
  - g. Your issue types should look like this:



- 4. Add fields to an issue type:
  - a. Click the **Bug** issue type.
  - b. On the right, in **Search all fields** search for **Priority** and select it. It's placed in the Description fields area and above the HIDE WHEN EMPTY line.
    - Fields below the HIDE WHEN EMPTY line remain hidden in the issue if they don't contain a value. Once they have a value they appear on the issue. You can always show hidden fields by clicking Show more on the issue.
- 5. Grab the new **Priority** field and drag it around. Place it between Status and Assignee.
- 6. Create a new custom field to track platform. On the right, click Checkbox:
  - a. Field name: **Platform**
  - b. Optional description: Choose which platform the bug appears on
  - c. Highlight Option 1 text and replace it with Android
  - d. Add another option, iOS
  - e. Collapse **Platform** field details (click down arrow).
  - f. Drag the new **Platform** field to beneath **Reporter** in the **HIDE WHEN EMPTY** section.

#### g. Click Save changes.



**Note**: The Platform custom field is available and can be reused in other issue types in this team-managed project.

- 7. Test the new issue type:
  - a. Create a new issue in the **Galaxy App** project of type **Bug**:
    - i. Add a Summary e.g. Planets are not showing.
    - ii. Optional Description: **Planets don't show on the Find a Planet page in the app**.
    - iii. Scroll down and ensure you see the **Priority** field. Change it if you wish.
    - iv. Scroll down and ensure you see the **Platform** field. Leave both options unchecked.
    - v. Click Create.
  - b. Open your new bug issue.
    - Question: Do you see the Platform field? If not, why not?
    - Answer: You don't see the Platform field because it doesn't have a value.
- 8. Click the **More fields** dropdown.
  - a. Click the Platform field value 'None' and select both Android and iOS.
  - b. You should still see the field as it now has values.
  - c. Refresh your page and confirm you see the field in your main details panel.



### Exercise 3 Configuring the Board & Customizing Features

- 1. Add a new column to the project board:
  - a. Click **Board** in the project sidebar.
  - b. Click the < at the top of the project sidebar or enter [ to collapse the project sidebar.
  - c. Add a review step in their work process (workflow). Click + to the right of the Done column.
    - i. Name the new column **Review**.
    - ii. Click **Return** or the tick sign.
  - d. Grab the top of the column and drag it to just before DONE.
- 2. Customize project features:
  - a. Expand the project sidebar and note the options in the sidebar.
  - b. Go to the project settings **Features** page.
    - Note: Max's team has grown, and they want to start using scrum with sprints instead of Kanban. They also want to track their performance with reports.
  - c. Select **Backlog**, **Sprints**, **Reports**, and **Estimation**.
  - d. Return to the project and note the project sidebar now has Backlog and Reports.
  - e. Go to the **Backlog** and you now see an area to create sprints at the top of the page.

**Note:** With team-managed projects, you can customize the features in your projects regardless of what type of project template you started with.

Congratulations on completing the lab!

## **Appendix**

### Further reading

Reference	URL
Manage and administer team-managed projects	https://support.atlassian.com/jira- software-cloud/docs/manage-and- administer-team-managed-projects/
Overview of permissions in team-managed projects	https://support.atlassian.com/jira- software-cloud/docs/next-gen- permissions/
Customize an issue's fields in team-managed projects	https://support.atlassian.com/jira- software-cloud/docs/customize-an-issues- fields-in-team-managed-projects/

### Best practices

Pitfall	Example Use Case	Best Practice
Creating a long difficult to remember project key results in users getting frustrated because they cannot easily search for issues.	You created the key HMN_RSS for the new Human Resources project but users of this project are complaining because they can't remember the key to easily find issues.	You can change the default project key but if you do, choose a key that's descriptive and easy to type. Users often use the issue key to find issues and you want to make it as easy as possible for them. Don't use overly long keys.
Screens that contain too many fields will make users	Users are frustrated with filling out a long Create Issue screen as there are too many fields and they don't know what half of them are. So, they just	Remove unused fields to help simplify the screens for users and make them easier to use.

Pitfall	Example Use Case	Best Practice
frustrated and they'll skip filling out important fields.	leave most of them blank and you miss out on getting data on some of the important fields, so gadgets and reports don't work well.	