

 **ATLASSIAN** University

Managing Jira Projects Data Center

Lab Workbook

 **ATLASSIAN** University

Table of Contents

Managing Jira Projects Introduction	1
Lab 1 Course Overview	2
Logging in to Confluence or Jira:.....	7
Lab 2 Managing Projects.....	9
Exercise 1 Creating a Project.....	10
Appendix.....	11
Lab 3 Managing Roles & Permissions	12
Exercise 1 Managing Roles & Permissions	13
Appendix.....	17
Lab 4 Managing Boards	18
Exercise 1 Adding a New Column	19
Exercise 2 Editing Workflows.....	20
Exercise 3 Modifying the Board Filter.....	23
Exercise 4 Managing Releases and Versions.....	26
Appendix.....	29
Lab 5 Boards & Projects	31
Pre-requisites	32
Exercise 1 Creating Multiple Boards in a Project	33
Exercise 2 Showing Multiple Projects on a Board	37
Appendix.....	38
Lab 6 Managing Issues.....	40
Exercise 1 Configuring Issue Types.....	41
Exercise 2 Configuring Components.....	42
Exercise 3 Using Bulk Change to Correct Labels	44
Exercise 4 Using Bulk Change to Move Issues from One Board to Another.....	45
Exercise 5 Managing Screens & Fields.....	49

Appendix.....	52
Lab 7 Reports & Dashboards	53
Exercise 1 Creating a Dashboard.....	54
Exercise 2 Running Reports	57
Appendix.....	60
Lab 8 Other Jira Features.....	62
Exercise 1 Creating a Project with Shared Settings.....	63
Exercise 2 Running Parallel Sprints	65
Appendix.....	66

Managing Jira Projects Introduction

Lab format

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 are also appendices, which you don't need to complete the course. They are full of useful information like additional reading and best practices. Dig into these after you complete the course!

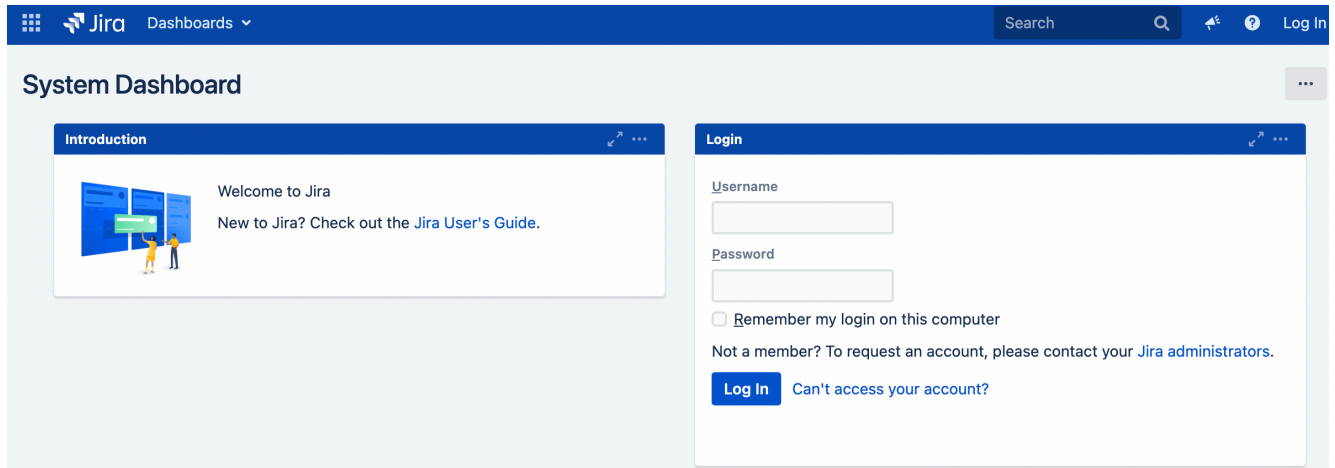
Logging in to your lab environment

To log in during the labs, you need the site URL that you've been assigned and each user's username and password. We'll cover accessing your site in the next lab.

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

You'll log in using the user or users listed below.

Name	Role
Dakota Jones	Jira administrator
Max Taylor	Project administrator
Cassie Owens	Scrum master
Ryan Lee	Jira user



The screenshot shows the Jira System Dashboard. The top navigation bar includes the Jira logo, 'Dashboards' dropdown, a search bar, and a 'Log In' button. The main content area is titled 'System Dashboard' and contains two side-by-side panels. The left panel, titled 'Introduction', features a welcome message and a link to the 'Jira User's Guide'. The right panel, titled 'Login', contains input fields for 'Username' and 'Password', a 'Remember my login on this computer' checkbox, and a 'Log In' button. A link for 'Can't access your account?' is also present.

System Dashboard

Introduction

Welcome to Jira

New to Jira? Check out the [Jira User's Guide](#).

Login

Username

Password

☐ Remember my login on this computer

Not a member? To request an account, please contact your [Jira administrators](#).

Log In [Can't access your account?](#)

9 Screenshot: log into Jira (or Confluence) using the username listed for the courses and the password Charlie!

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. Create a new scrum project:
 - a. Log in to Jira as the Jira administrator, Dakota Jones (username: **djones** password: **Charlie!**).
 - b. From the Jira menu click **Projects > Create project**.
 - c. On the **Create project** screen, scroll up and select the **Scrum Software Development** template.
 - d. Click **Next**, then in the next window that displays Issue Types and Workflow for the project, click **Select**.
 - e. Name the project **Megastorm** and leave the key at **MEG**.
 - f. Change the project lead to Max Taylor.
 - g. Click **Submit**
 - **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:** Choose a project 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 one.
2. Assign the project administrator role:
 - a. At the bottom of the project sidebar, click **Project settings** (the cog icon).
 - b. In the project settings sidebar, click **Users and roles**.
3. On the **Users and roles** page, click **Add users to a role**.
 - a. In the **Add users to a role** pop-up:
 - i. Start typing **max** in the Users or groups field and select **Max Taylor**.
 - ii. For role, select **Administrators**.
 - iii. Click **Add**.
 - b. Max is now listed as an administrator in the project.

Congratulations on completing the lab!

Appendix

Further Reading

Reference	URL
Start a new project	https://confluence.atlassian.com/jirasoftwareserver/starting-a-new-project-938845217.html

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 DEVLMT.	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.

Exercise 1 Managing Roles & Permissions

Change permissions for a project:

1. Log in to Jira as the Jira administrator, **Dakota Jones**.
 - a. Click **Projects**.
 - b. Select **View All Projects**.
 - c. Go to the **Megastorm** project.
 - d. In the sidebar, click the cog icon to go to the **Project settings** page.



- e. In the project settings sidebar, click **Permissions**.
 - **Note:** This project uses the Default software scheme which new Software projects use by default. A permission scheme associates permissions with users, groups, or project roles.
 - **Note:** Many permissions use Application access (Any logged in user).
- f. 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.
- g. View the **Delete Issues** permission.
 - **Question:** Who can delete issues in this project?
 - **Answer:** Project administrators can delete issues.
- h. At the top of the page, click **Actions** and select **Use a different scheme**.
 - i. 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.
 - **Question:** Now who can delete issues in this project?
 - **Answer:** The project administrator and members of the Issue managers project role can delete issues.

View project permissions as a project administrator:

1. Log in to your site as the Megastorm project administrator, **Max Taylor (mtaylor/Charlie!)**.
 - a. Go to the **Megastorm** project.
 - b. In the project sidebar, click **Project settings**.
 - c. 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.

Assign team members to project roles:

1. Assign Scrum masters and Issue managers custom project roles:
 - **Note:** In this project, Cassie Owens is the scrum master. Max is going to assign Cassie to the Project Lead, board administrator, and Scrum masters and Issue managers roles in the project. He will also assign Ryan Lee to the Issue managers role.
 - i. In the project settings sidebar, click **Users and roles**.
 - ii. Click **Add users to a role**.
 - iii. Start typing **Cassie** and select **Cassie Owens**.
 - iv. For Role, select **Scrum masters**.
 - v. Click **Add**.
 - vi. Repeat these steps to add **Cassie Owens** and **Ryan Lee** to the **Issue managers** role.
 - **Note:** If a user has already been assigned a role in a project, you can use the drop-down menu in the **Roles** column to assign additional roles.

Name	Email address	Roles
jira-administrators <small>jira-administrators</small>	--	Administrators
jira-developers <small>jira-developers</small>	--	Developers
jira-users <small>jira-users</small>	--	Users
Max Taylor <small>mtaylor@teams-in-space.com</small>	mtaylor@teams-in-space.com	Administrators
Cassie Owens <small>cowens@teams-in-space.com</small>	cowens@teams-in-space.com	Scrum masters

Assign project lead:

1. In the Users and roles page, click **Edit defaults**.
 - a. For Project lead, start typing **Cassie** and select **Cassie Owens**.
 - b. Set the Default Assignee as **Unassigned**.
 - c. Click **Update**.

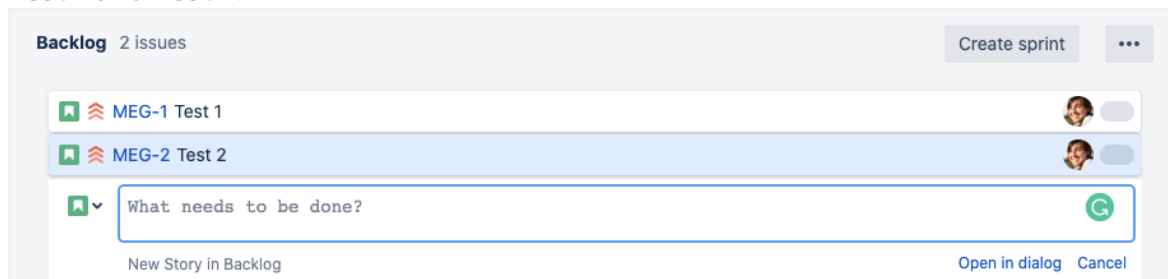
Assign board administrators:

1. At the top of the project sidebar, click **Active sprints**.
 - a. On the Active sprints page, click **Board > Configure**.
 - b. Go to the **General** page (if you're not already there).
 - **Question:** Who is currently listed in Administrators and why?
 - **Answer:** Dakota Jones is a board administrator because she created the board (it was automatically created as part of creating a Software Scrum

- project). Max Taylor is a board administrator because he was assigned the Project Lead role when the project was created.
- Mouse over the **Administrators** field and click the **edit icon**.
 - In the **Administrators** field, Start typing **Cassie** and select **Cassie Owens**.
 - Click outside the **Administrators** field to exit edit mode.
 - **Note:** Now Cassie, Dakota, and Max can configure the project's default board.

Verify your changes as the scrum master:

- Log in to your site as **Cassie Owens (cowens/Charlie!)**.
 - Go to the **Megastorm** project.
 - In the project sidebar, click **Backlog**.
 - On the Backlog page, create two issues (of any type) and give them any name, e.g. Test 1 and Test 2.

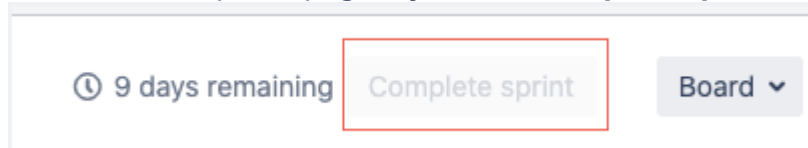


- Click **Create sprint**.
 - Accept the defaults and click **Create**.
 - Drag the two issues up into the sprint and click **Start sprint**.
 - Accept the defaults and click **Start**.
 - **Note:** Now you see the sprint board with the two issues. This confirms Cassie has the Manage sprints permission.
- Click one of the issues, then:
 - Issue details will appear in the right-side area.
 - Click the three dot menu.



- Confirm you see the **Delete** option (but don't delete it).
 - **Note:** This confirms Cassie has the Delete Issues permission (because she has the Issue managers role).

- f. Access the **Board settings**.
 - **Note:** This confirms Cassie is a board administrator for the project's default board.
- g. Log out as Cassie.
2. Verify your changes as a developer:
 - a. Log in to your site as **Ryan Lee (rlee, Charlie!)**.
 - b. Go to the **Megastorm** project.
 - c. On the Active sprints page, try to click **Complete sprint**.



- **Question:** Was Ryan able to complete the sprint? If not, why?
 - **Answer:** No, Ryan cannot complete the sprint, that option is greyed out. This is because he doesn't have any of the Sprints permissions (**Edit Sprints, Manage Sprints, Start/Complete Sprints**).
- d. From the sprint board, click an issue to see its details on the right side. Click the three dot menu and confirm he has the option to delete the issue (but don't delete it).
 - **Note:** This confirms Ryan has the Delete Issues permission.
 - e. Click the **Board** menu and select **Configure**.
 - **Question:** Can Ryan configure the board? For example, on the General page, can he edit the Administrators? If not, why?
 - **Answer:** No, Ryan cannot configure the board as he's not a board administrator. Note the message at the top telling him to contact a Jira or Board Administrator to configure this board.
 - f. Log out as Ryan.

Congratulations on completing the lab!

Appendix

Further Reading

Reference	URL
Permissions overview	https://confluence.atlassian.com/jirasoftwareserver/permissions-overview-939938996.html

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 the first lab in this course, you can use the existing Megastorm project. If you didn't create this project, use the instructions in [Lab 2 Exercise 1](#) to create the Megastorm project.

Exercise 1 Adding a New Column

① 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.

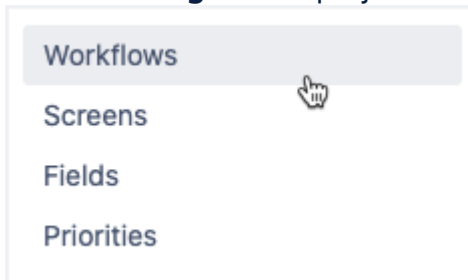
1. Log in to Jira as the project administrator, **Max Taylor (mtaylor/Charlie!)**.
2. Go to the **Megastorm** project's **Active sprints** page.
3. Click the **Board** menu on the top right and select **Configure**.
4. On the board configuration sidebar, click **Columns**.
5. Click **Add column**:
 - a. Name: **QA**
 - b. Category: **In Progress**
 - c. Click **Add**.
 - **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. Max was able to do this because he is a project administrator.
6. Grab your new **QA** column and move it before In Progress then move it back 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 any issue type, enter any Summary, and hit **Return**.
 - d. Drag the issue from the backlog into the sprint and, if prompted, click **Confirm**.
9. Return to the **Active sprints** and click an issue to display the issue details in the right panel.
10. Under detail, locate the issue's **Status** field. Click the **View Workflow** link. This opens the issue.
 - a. Click the **View Workflow** link in the issue. This will open up a non-editable diagram of the workflow used for this issue.
 - b. **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.
11. Close the workflow dialog.

Exercise 2 Editing Workflows

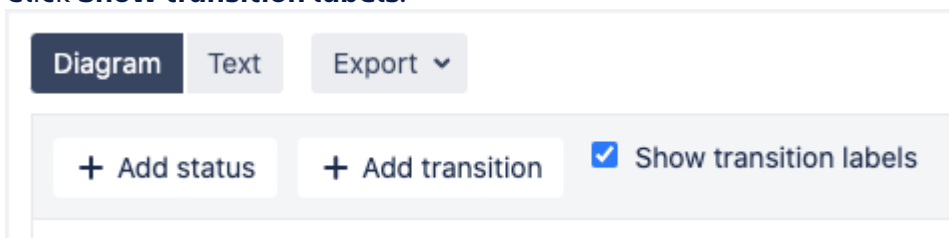
- ① Max's team has decided that:
- Issues should only be able to move to QA from In Progress.
 - Issues that are in QA should be able to move to any other status.
- In this exercise, you use the workflow editor to make these changes.

View the workflow editor.

1. Log into Jira as the Megastorm project administrator, **Max Taylor (mtaylor/Charlie!)**.
2. Go to the **Megastorm** project.
3. Go to the **Megastorm** project settings **Workflow** page.



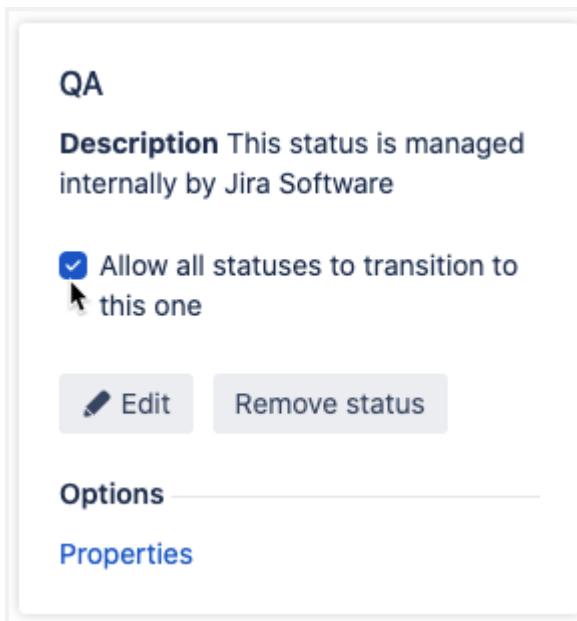
4. Click **Software Simplified Workflow for Project MEG**.
5. Click the **Edit** button.
 - **Note:** Project administrators can only edit workflows that are not shared with any other projects and in projects where the Extended project administration permission is granted.
 - **Note:** You're editing a draft of the workflow. Live workflows cannot be edited.
6. Click **Show transition labels**.



Add a transition to your project workflow

① Transitions can utilize triggers, conditions, validators, and post functions. They can use transition screens, which prompt the user to update fields or add comments as the transition is being executed. Only the Jira administrator can create and edit these advanced workflow elements.

1. In the workflow editor, drag the statuses so they're in the order they'd typically be executed in (TO DO, IN PROGRESS, QA, DONE).
 - **Note:** We need to remove the global transition for QA that allows all statuses to transition to this status.
2. Click the **QA** status and in the QA dialog, uncheck **Allow all statuses to transition to this one**.

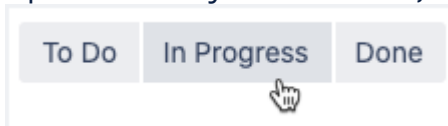


3. In the QA dialog, click **Add transition** and enter:
 - a. From status: **In Progress**
 - b. To status: **QA**
 - c. Name: **QA**
 - d. Optional Description: **Moving an issue from being worked on to in QA.**
4. Click **Add**.
5. Click any whitespace in your workflow editor to close the Review transition window.
6. Click **Publish** to save the workflow and make it active.

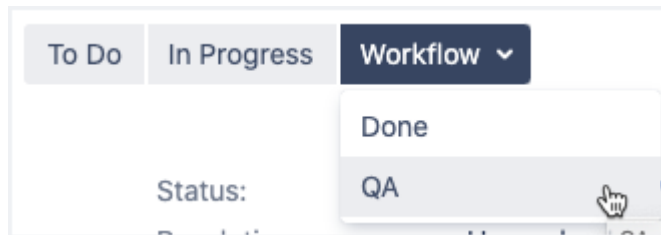
Test the updated workflow

- ① You should always test a workflow immediately after editing it to confirm your changes are working as expected. You can test the workflow by walking test issues through the workflow.

1. Click **Create** and create a new Story issue in the Megastorm project. Give the issue any name such as **Issue Workflow Test**.
2. Open the newly created issue, and click the **In Progress** transition button.



- **Note:** Now you should see a Workflow dropdown with both Done and QA transitions available.



3. Click the **QA** transition and confirm that status was updated to QA.
4. Click the **Done** transition
5. Confirm you only see the **To Do**, **In Progress**, and **Done** transitions.
 - **Note:** If the workflow has been configured as expected, the issue cannot transition to QA.
6. Go to the project's Active board and test the workflow on the board issues by dragging them in and out of the QA column.
7. Confirm you can only move an issue to QA from the In Progress column.
8. Confirm you can move an issue from QA to any other column.

View the impact of changing a simplified workflow

1. From the Megastorm project board, return to board column configuration (**Board > Configure > Columns**).
 - **Question:** Can you add a new status to the board?
 - **Answer:** No, you can no longer add new statuses to the board. You can only add statuses if the board uses the simplified workflow. Once you edit the workflow, it's no longer using the default simplified workflow. If you want to add any more statuses, you must do so from the workflow editor.

Exercise 3 Modifying the Board Filter

i 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.

The owner of a filter, along with any filter editors, can edit the filter. Since the Jira administrator (Dakota Jones) created the project, and thereby the board, Dakota owns the filter for the default board. To allow the project administrator to edit the board filter, there are two options:

1. the Jira administrator can grant Max ownership of the filter.
2. the owner of the filter can add Max as a filter editor.

In this exercise, Dakota will add Max as a filter editor so he can manage the board's filter. Max will then update the board's filter to exclude bugs from the board.

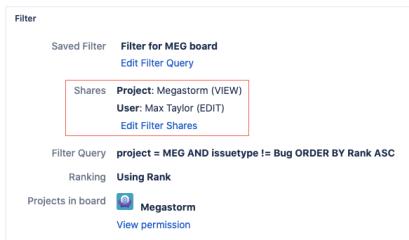
Manage the board filter:

1. Log into Jira as the Jira administrator, **Dakota Jones**.
2. Go to the **Megastorm** project's **Active sprints** page.
3. Click the Board menu and select **Configure**.
4. On the board configuration sidebar, click **General**.
5. In Shares, click **Edit Filter Shares**.

6. From the **Add Editors** drop-down menu, select **User**.
7. In the User field, start typing **max** and select **Max Taylor**.
8. Click **+Add**. **Note:** This is an easy step to miss, so be careful!

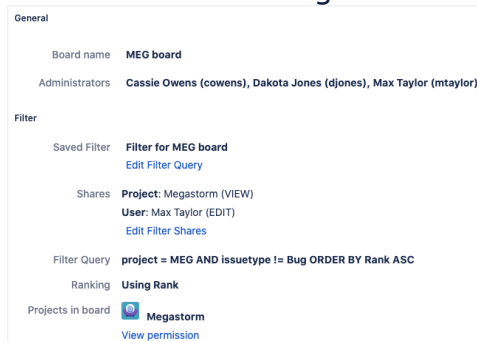
9. Click **Save**.
10. Return to the **Megastorm** project's **Active sprints** page.

11. Click the **Board** menu and select **Configure**.
12. Max should now be listed in the filter shares for the MEG board.



Modify the board filter:

1. Log in to your site as the project administrator, **Max Taylor**.
 - a. Go to the **Megastorm** project's **Active sprints** page.
 - b. 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.
 - c. Click the Board menu and select **Configure**.
 - d. On the board configuration sidebar, click **General**.
 - e. 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.
 - f. If you don't see the JQL query, click **Advanced** in order to switch to JQL.
 - g. Update the JQL query by adding the highlighted part.
project = MEG **AND issuetype != Bug** ORDER BY Rank ASC
 - h. Click **Search** and you now see all non-bug issues.
 - i. Click **Save** next to the name of the filter. You should see a confirmation that the filter was saved successfully.
 - j. Return to the Megastorm project's **Active sprints** page. You shouldn't see any bugs shown anywhere on the board, either here or in the Backlog.
 - k. 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 in the backlog view will reflect only counts of issues that match the board filter.

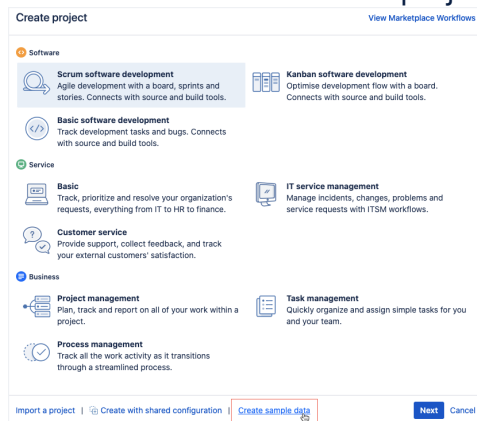
3. Issues created in the project which are not in the board's filter won't be visible on the board. Users will see a message that the issue was created but it is not currently visible.

Exercise 4 Managing Releases and Versions

i In this exercise, you create a new project with sample data which will also create releases and versions. Only Jira administrators can create projects.

Create a new project with sample data:

1. Log into Jira as the Jira administrator, **Dakota Jones**.
 - a. From the **Projects** menu, select **Create project**.
 - b. Select the **Scrum software development** template.
 - c. At the bottom of the Create project page, click **Create sample data**.



- d. Select the **Scrum software development** template.
- e. Click **Next**.
 - i. Name the project **Caprica**.
 - ii. Use the default project key **CAP**.
 - iii. Add Max as the **Project Lead**.
 - iv. Click **Submit**.
 - v. After a short wait, you should see your new project's backlog which is full of issues.
 - vi. Set **Max** as the **project administrator**:
 1. At the bottom of the project sidebar, click **Project settings** (or the cog icon).
 2. In the project settings sidebar, click **Users and roles**.
 3. On the **Users and roles** page, click **Add users to a role**.
 4. In the Users or groups field, start typing **max** and select **Max Taylor**.
 5. In the role field, select **Administrators**.
 6. Click **Add**.

View Releases and Versions:

1. Go to your site URL and log in as the project administrator, **Max Taylor**.
 - a. Go to the **Caprica** project.
 - b. In the project sidebar, click **Releases** (the ship icon).
 - **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.
 - c. Click the **Released** button above the versions to 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 **Archived** button. There are no archived versions in this project yet.
 - d. 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.
 - e. 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.
 - f. Click one of the issues to open it.
 - **Note:** Note the Fix Versions fields. Once a version has been created for a project, this field becomes available for your issues.
 - g. 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.

Release a version:

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

1. Either click Back in your browser or return to the project's **Releases** page then click **Version 2.0**.
 - a. Click **Release**:
 - i. Check **Move issues to version** then select **Version 3.0**.
 - ii. For Release date, enter today's date.
 - iii. Click **Release**.
 - b. Return to the **Releases** page and click the **Released** button. The status of Version 2.0 is now **Released**.
 - c. Click ... (three dots) for Version 2.0.

- **Note:** To revert the release of a version, you can unrelease the version here. You can also release unreleased versions here. Leave it as is.

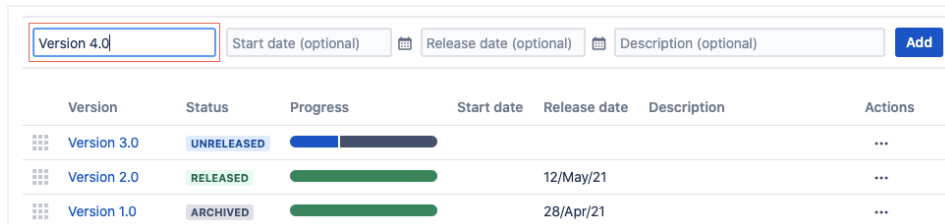
Archive a version:

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.

1. From the **Releases** page, click ... (three dots) next to Version 1.0 and select **Archive**.
 - a. Select **Archived** from the drop-down and confirm status is now **ARCHIVED**.

Add a new version:

1. On the **Releases** page, click in the **Version name** field to create a new version.
 - a. Name: **Version 4.0**.



The screenshot shows the 'Add' button in the top right corner of the 'Releases' page. Below it is a table with the following columns: Version, Status, Progress, Start date, Release date, Description, and Actions. The table contains three rows: Version 3.0 (UNRELEASED), Version 2.0 (RELEASED), and Version 1.0 (ARCHIVED). The 'Version 4.0' text is entered in the 'Version name' field above the table.

Version	Status	Progress	Start date	Release date	Description	Actions
Version 3.0	UNRELEASED	<div></div>				...
Version 2.0	RELEASED	<div></div>		12/May/21		...
Version 1.0	ARCHIVED	<div></div>		28/Apr/21		...

- b. Start Date: Enter today's date.
- c. Release date: a date in the future.
- d. Click **Add**.
- e. 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.
- f. Cancel out of issue creation.
- g. 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.
- h. Go to project settings and click **Versions**.
 - **Note:** You can perform all the same release tasks here. The Releases page in the project enables team members to see the status of versions.

Congratulations on completing the lab!

Appendix

Further Reading

Reference	URL
Configuring a board	https://confluence.atlassian.com/jirasoftwareserver/configuring-a-board-938845252.html
Managing shared filters	https://confluence.atlassian.com/adminjiraserver073/managing-shared-filters-861254012.html
Releases and versions	https://confluence.atlassian.com/jirasoftwareserver/planning-a-version-938845382.html

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	Your team has 50 issues on the board. They can't quickly	Create quick filters so your team can quickly see important issues. Update the board filter to permanently

too many issues on the board.	see the high priority issues or ones that matter to them.	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. In order to complete the lab we must first create two projects to use for our shared board.

Pre-requisites

Create two new projects:

1. Log in to Jira as the Jira administrator, **Dakota Jones**.
2. Click **Projects** in the top navigation bar and select **Create project**.
3. In the **Create project** window, select the **Kanban software development template**.
 - a. Click **Next**.
 - b. Click **Select** to confirm the **Kanban software development** template.
 - c. Name the project **Enterprise**.
 - d. Add the project key **ENT**.
 - e. Add **Max Taylor** as the Project lead.
 - f. Click **Submit**.
4. At the bottom of the project sidebar, click **Project settings** (or the cog icon).
5. In the project settings sidebar, click **Users and roles**.
6. On the Users and roles page, click **Add Users to a role**.
 - a. In the Users or groups field, start typing **max** and select **Max Taylor**.
 - b. In the Role field, select **Administrators**.
7. Click **Add**.
8. Repeat these steps to create a Kanban project called **Voyager**.
 - **Note:** Be sure to add Max Taylor as the **Project Lead** when you create the board. This will automatically add him as the project's board administrator. Then add him to the project administrator role.

Change the owner of the **Filter for Voyager board**:

 **Note:** You can use the . (period) **shortcut** and type Filters into the drop-down menu.

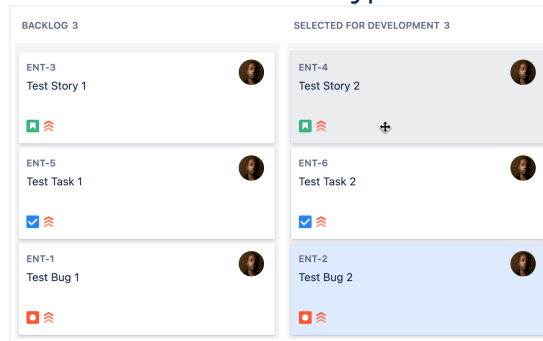
1. Click the administration cog icon in the Jira menu. From the drop-down **select** System.
 - a. Under **SHARED ITEMS** in the sidebar click **Filters**.
 - b. In the **Filter for Voyager board** row, click the cog icon and change the owner to **Max Taylor**.

Exercise 1 Creating Multiple Boards in a Project

- i** 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.

View the project's default board:

1. Log in to Jira as the project administrator, **Max Taylor**.
 - a. Go to the **Enterprise** project's **Kanban board**.
 - b. If you just created this project:
 - i. Create 2 bugs, 2 stories, and 2 tasks. Give them simple summaries such as Test bug 1.
 - ii. Place one of each issue type in the Backlog column.
 - iii. Place one of each issue type in the Selected for Development column.



- **Note:** This default kanban board shows all the issues in the project. We'll use this as our primary board.

Create a scrum board for stories:

1. At the top of the page, click the board drop-down and select **Create board**.
 - a. Click **Create a Scrum board**.
 - b. Ensure **Board from an existing project** is checked and click **Next**.
 - c. On the Name this board dialog:
 - i. Board name: **Story board**
 - ii. Project: **Enterprise**
 - iii. 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.

Create a sprint:

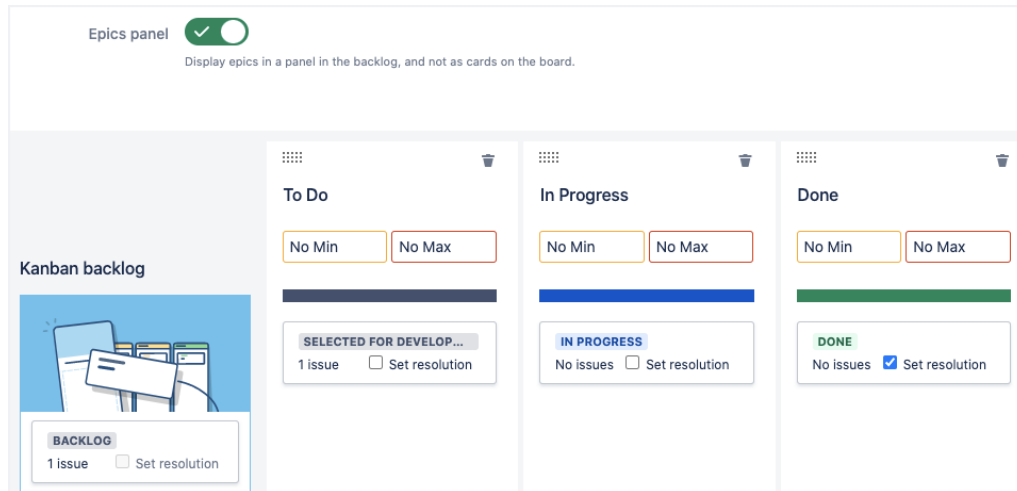
1. From the **Backlog**, click **Create sprint**.
2. In the **Create sprint** dialog, accept the defaults and click **Create**.
 - a. Drag one issue of each type into the sprint.
 - b. Click **Start sprint**.
 - c. On the Start sprint dialog, accept the defaults and click **Start**.
 - **Note:** For this board we only want to see stories and tasks.
 - d. On the Active sprints page, click the **Board** menu and select **Configure**.
 - i. On the General page, click **Edit Filter Query**.
 - ii. If you see the JQL query, click **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.
 - e. Return to the **Enterprise** project.
 - f. View the Story board's **Active sprints** and **Backlog** pages. On both, you should only see stories and tasks now.

Create kanban board for bugs:

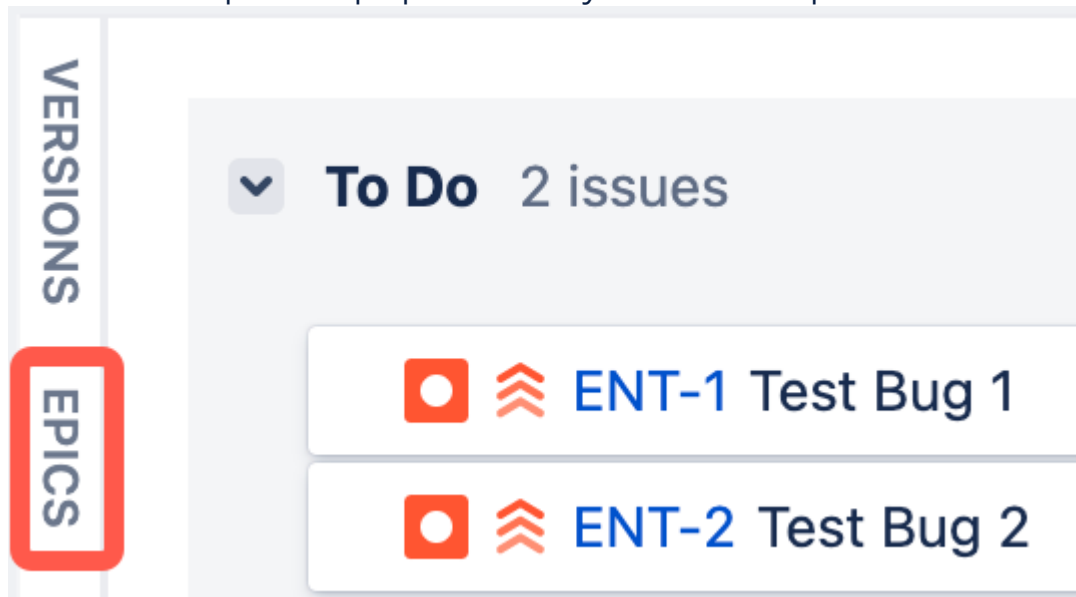
1. In the Enterprise Project, click the board drop-down and select **Create board**.
 - a. Click **Create a Kanban board**.
 - b. Ensure **Board from an existing project** is checked and click **Next**.
 - c. On the Name this board dialog:
 - i. Board name: **Bug board**
 - ii. Project: **Enterprise**
 - iii. 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.
 - d. On the Bug board, **Configure** the board:
 - i. On the General page, click **Edit Filter Query**.
 - ii. If you see the JQL query, click **Basic**.
 - iii. Click **Type** drop-down and select the **Bug** issue type.
 - iv. Next to the filter name, click **Save**.
 - e. 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.

Enable kanban backlog:

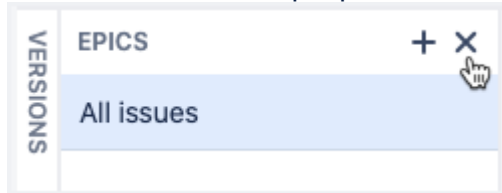
1. On the Bug board, **Configure** the board.
 - a. Go to the **Columns** page.
 - b. Drag the **Backlog** status into the Kanban backlog to the left.
 - c. Ensure there is a green checkmark next to the **Epics panel**.



- d. Click **Back to board**. Now you only see one bug as the other bug was moved to the new backlog.
- e. 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 manage 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.
- f. Drag a bug from **Backlog** to **To Do**.
- g. Click **EPICS** to open the epic panel where you can create epics.



- h. Click **x** to close the epic panel.



- i. In the project sidebar, click **Kanban board** to return to the Bug board and now you see both bugs in the To Do column.

View primary board:

1. In the project sidebar, click the **board drop-down** and select **ENT 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. 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

① Max is managing two projects that have merged and he wants to see all the issues on one board.

View multiple projects on a single board:

1. Log into Jira as the project administrator, **Max Taylor**.
 - a. Go to **Voyager** project's **Kanban board**.
 - i. Create 2 new issues of any type. Give them simple summaries such as Test bug 1.
 - ii. Leave them both in the Backlog column.
 - **Note:** This default kanban board shows all the issues in the project.
 - b. Click the **Board** dropdown and select **Configure**.
 - i. On the **General** page, click **Edit Filter Query**.
 - ii. If you see the JQL query, click **Basic**.
 - iii. Click the project drop-down (first on left). Leave Voyager selected and also select **Enterprise**.
 - iv. Next to the Filter Name, click **Save**.
 - **Note:** The Jira administrator (Dakota) who created the project, and who originally owned this board filter, transferred ownership to Max as part of the [pre-requisite tasks for this lab](#).
 - c. 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.

Configure swimlanes by project:

To make it easier to differentiate which issues come from which project, we set up swimlanes per project.

1. Click the **Board** dropdown and select **Configure**.
 - a. On the board configuration sidebar, click **Swimlanes**.
 - b. Click the **Base Swimlanes on** drop-down and select **Projects**.
 - c. 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?	https://confluence.atlassian.com/jirasoftwareserver/creating-a-board-938845220.html#Creatingaboard-Beforeyoubegin
Create a board	https://confluence.atlassian.com/jirasoftwareserver/creating-a-board-938845220.html
Use your scrum backlog	https://confluence.atlassian.com/jirasoftwareserver/using-your-scrum-backlog-938845361.html

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, use the instructions in [Lab 2 Exercise 1](#) to create the Megastorm project.

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.

Create a new issue type:





1. Log into Jira as the Jira administrator, **Dakota Jones**.
 - a. Go to the **Megastorm** project.
 - b. 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.
 - c. Click **MEG: Scrum Issue Type Scheme** which takes you to the Issue types page in the project.
 - d. Click **Actions – Edit issue types**.
 - **Note:** This takes you to the Modify Issue Type Scheme page for the project in Jira administration.
 - e. 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**.
 - f. 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. **Note** that when a user selects an issue, this will be presented as the default **unless** you have already created an issue in that project. If you have already created a bug, for example, then the default reverts to the bug issue type, regardless of your settings in the Issue Type Scheme.
 - g. Click **Save**.
 - h. Confirm your new Feature Request issue type is now listed in the current issue type scheme for the Megastorm project.

Exercise 2 Configuring Components

- ① 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.

Create components:

1. Log into Jira as the project administrator, **Max Taylor**.
 - a. Go to the **Megastorm** project settings **Components** page.
 - b. Click in the **Component Name** field and enter:
 - i. Name: **Database**
 - ii. Lead: **Max Taylor**
 - iii. Default Assignee: **Component Lead (Max Taylor)**
 - iv. Click **Add**
 - c. Click the ... (three dots) in the row for the new component. You can edit or delete the component here. Don't do that now.
 - d. Create four more components:
 - **1. Name: User Interface**
 - Lead: **Empty**
 - Default Assignee: **Component lead (Unassigned)**
 - **2. Name: Localization**
 - Lead: **Max Taylor**
 - Default Assignee: **Project lead**
 - **3. Name: Login**
 - Lead: **Cassie Owens**
 - Default Assignee: **Project default**
 - **4. Name: Storm Data**
 - Lead: **Cassie Owens**
 - Default Assignee: **Unassigned**

Component	Status	Issues	Lead	Description	Default assignee
User Interface	ACTIVE	0 Issues			Component lead
Database	ACTIVE	0 Issues	 Max Taylor		Component lead
Localization	ACTIVE	0 Issues	 Max Taylor		Project lead
Login	ACTIVE	0 Issues	 Cassie Owens		Project default
Storm Data	ACTIVE	0 Issues	 Cassie Owens		Unassigned

Test auto-assignment of components:

1. 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, Megastorm's project lead is set as the default assignee, so the issue is assigned to Cassie Owens.
- **Question:** If you created an issue and selected the **Login** component, who would the issue be assigned to?
- **Answer:** Even though Cassie Owen 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 Cassie Owen 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, there is no component lead defined for this component, so the new issue is set to Unassigned.
- **Question:** If you created an issue and selected the **Database** component and set the Assignee to Cassie Owen, would the issue stay assigned to Cassie 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 auto-assignment. It would stay assigned to Cassie Owen.
- **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. Automatic assignment takes place when components are added when an issue is created, not at other stages of the issue lifecycle.

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 issues 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 **Advanced** to go to advanced search.
 - c. Enter the JQL:

project = Megastorm 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 **Tools menu** 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 **Confirm**.
 - g. When the bulk operation is 100% complete, click **OK, got it**.
 - 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.

Set up the project to move issues from:

1. Log into Jira as the Jira administrator, **Dakota Jones**.
 - a. From the Jira menu click **Projects > Create project**.
 - b. On the **Create project** screen, select the **Basic software development** template.
 - c. Click **Next**, then in the next window that displays Issue Types and Workflow for the project, click **Select**.
 - d. Name the project **Hurricanes** and leave the key at **HUR**.
 - e. Enter **Max Taylor** as the Project Lead.
 - f. Click **Submit**.
 - g. On the **Users and roles** page, click **Add users to a role**.
 - i. In the **Add users to a role** pop-up:
 1. Start typing **max** in the Users or groups field and select **Max Taylor**.
 2. For role, select **Administrators**.
 3. Click **Add**.
 - ii. Max is now listed as an administrator in the project.

View project issue types:

1. In the **Hurricanes** project, go to the **Issue types** project settings page.
 - a. Note the issue types for this Software Bug tracking project – **Bug, Epic, Improvement, New Feature, Task, and Sub-task**.
 - b. Go to the **Megastorm** project and view the **Issue types** project settings page.
 - c. 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 Basic software development 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 Basic software development project.

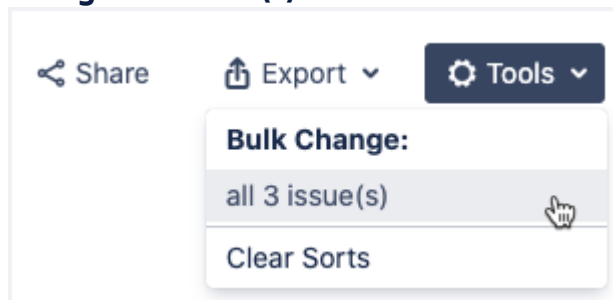
Create issues to move:

1. Go to your site URL and log in as the project administrator, **Max Taylor**.
 - a. Go to the **Hurricanes** project.

- b. Create three issues in the **Hurricanes** project.
 - i. Create a Task, a New Feature, and an Improvement.
- c. On the **Issues** page, refresh the page.

Bulk move the issues:

1. On the **Issues** page, Click **View all issues and filters** to go to the Search page.
 - a. Click the **Tools dropdown menu** on the top right of the search page and select **Bulk change all 3 issue(s)**.







- b. Click the checkbox on the left above the issues to select all the issues and click **Next**.
- c. Select **Move Issues** and click **Next**.
- d. 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**.
- e. View the Update Fields for Target Project 'Megastorm' – Issue Type 'Story' 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.
- f. Enter a Story Points field value of **3** and click **Next**.
- g. On the Update Fields for Target Project 'Megastorm' – Issue Type 'Task' page, click **Next**.
- h. On the Confirmation page, confirm that 1 Task, 1 New Feature, and 1 Improvement in the Hurricanes project will be affected. Click **Confirm**.



Megastorm — Story	
<i>i</i> The change will affect 2 issues with issue type(s) ↑ Improvement, + New Feature in project(s) Hurricanes .	
Issue Targets	
Target Project	Megastorm
Target Issue Type	↑ Story
Updated Fields	New Value
Story Points	3
Platform	
Business Value	
Megastorm — Task	
<i>i</i> The change will affect 1 issues with issue type(s) ✓ Task in project(s) Hurricanes .	
Issue Targets	
Target Project	Megastorm
Target Issue Type	✓ Task



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



Backlog 6 issues Create sprint ...



  MEG-7 tornado



  MEG-8 toranado



  MEG-9 tornad



  MEG-15 Hurricane task


  MEG-14 Hurricane improvement


  MEG-13 Hurricane new feature


 



 3

 3

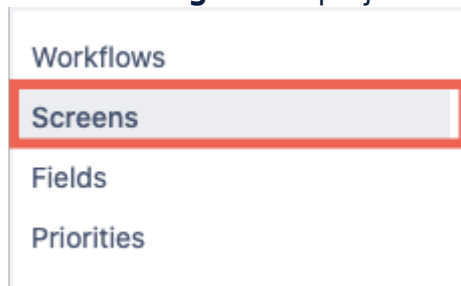
+ Create issue

Exercise 5 Managing Screens & Fields


- ① Your team decides that they need to use the platform field to gather data on where the bug impacts, so we'll add an existing field to the screen for the bug issue type in the Megastorm project.










Review the Megastorm project's screen configuration

1. Log into Jira as the Megastorm project administrator, **Max Taylor**.
 - a. Go to the **Megastorm** project.
 - b. Go to the **Megastorm** project settings **Screens** page.













- c. Click > **MEG: Scrum Default Screen Scheme**.
 - **Note:** Here you see the screen used for all Issue Types in the project, except the Bug Issue Type.

▼ MEG: Scrum Default Screen Scheme **DEFAULT** 

These 5 issue types...	...use this screen scheme								
<input checked="" type="checkbox"/> Story DEFAULT <input checked="" type="checkbox"/> Epic <input checked="" type="checkbox"/> feature requests <input checked="" type="checkbox"/> Task <input checked="" type="checkbox"/> Sub-task	<table border="1"> <thead> <tr> <th>Operation</th> <th>Screen</th> </tr> </thead> <tbody> <tr> <td>Create Issue</td> <td> MEG: Scrum Default Issue Screen</td> </tr> <tr> <td>Edit Issue</td> <td> MEG: Scrum Default Issue Screen</td> </tr> <tr> <td>View Issue</td> <td> MEG: Scrum Default Issue Screen</td> </tr> </tbody> </table>	Operation	Screen	Create Issue	 MEG: Scrum Default Issue Screen	Edit Issue	 MEG: Scrum Default Issue Screen	View Issue	 MEG: Scrum Default Issue Screen
Operation	Screen								
Create Issue	 MEG: Scrum Default Issue Screen								
Edit Issue	 MEG: Scrum Default Issue Screen								
View Issue	 MEG: Scrum Default Issue Screen								

- **Question:** Which screen is being used by the Create, Edit, and View operations?
 - **Answer:** The **Create**, **Edit**, and **View** operations use the same screen, the **MEG: Scrum Default Issue Screen**.
- d. Click > **MEG: Scrum Bug Screen Scheme**.
 - **Note:** Here you see the screen used for the Bug Issue Type.

MEG: Scrum Bug Screen Scheme									
This issue type...	...uses this screen scheme								
 Bug	<table> <tr> <th>Operation</th><th>Screen</th></tr> <tr> <td>Create Issue</td><td> MEG: Scrum Bug Screen</td></tr> <tr> <td>Edit Issue</td><td> MEG: Scrum Bug Screen</td></tr> <tr> <td>View Issue</td><td> MEG: Scrum Bug Screen</td></tr> </table>	Operation	Screen	Create Issue	 MEG: Scrum Bug Screen	Edit Issue	 MEG: Scrum Bug Screen	View Issue	 MEG: Scrum Bug Screen
Operation	Screen								
Create Issue	 MEG: Scrum Bug Screen								
Edit Issue	 MEG: Scrum Bug Screen								
View Issue	 MEG: Scrum Bug Screen								

- **Question:** Which screen is being used by the Create, Edit, and View operations?
- **Answer:** The Create, Edit, and View operations use the same screen, the **MEG: Scrum Bug Screen**.

Add a field to the screen used for the Bug Issue Type

1. Click **MEG: Scrum Bug Screen** (click any link in the Screen column for any Operation).
2. Click **Field name** at the bottom of the list of fields on the **Scrum Bug Screen**.
3. Start to type Platform to find the **Platform field** and then select it.

Add

Platform
Select List (single choice)

4. Click **Add**.

Position a field in the screen used for the Bug Issue Type

1. Click the **three lines** next to the **Platform field** in **MEG: Scrum Bug Screen**.
2. Drag the **Platform field** so that it sits under the **Environment field**.

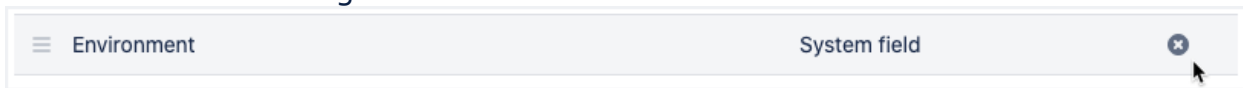
Confirm the Platform field has been added to the Bug Issue Type

1. Click **Create** and select Megastorm Project and Bug Issue Type.
2. Confirm that the Platform field appears in the issue and that it is positioned under the Environment field.
3. Click cancel to cancel out of the issue creation process.

Remove a field from the screen used for the Bug Issue Type

① Having both the Platform and Environment field on a screen could be confusing, so we'll remove the Environment field.

1. Go to the **Megastorm** project.
2. Go to the **Megastorm** project settings **Screens** page.
3. Click **> MEG: Scrum Bug Screen Scheme**.
4. Click **Meg: Scrum Bug Screen**.
5. Mouse over the Environment field.
6. Click the **x icon** on the right side of the field to remove it from the screen.



Confirm that the Environment field has been removed from the Bug Issue Type

1. Click **Create** and select **Megastorm Project** and **Bug Issue Type**.
2. Confirm that the **Environment field** does not appear in the issue.
3. Click **Cancel** to cancel out of the issue creation process.

Congratulations on completing the lab!

Appendix

Further Reading

Reference	URL
Adding, editing, and deleting an issue type	https://confluence.atlassian.com/jirasoftwareserver/creating-issues-and-sub-tasks-939938925.html
Organize work with components	https://confluence.atlassian.com/jirasoftwareserver/organizing-work-with-components-939938952.html
Edit multiple issues at the same time	https://confluence.atlassian.com/jirasoftwareserver/editing-multiple-issues-at-the-same-time-939938937.html

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 Reports & Dashboards



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

- ① First, you create a new project with sample data which creates a couple of sprints. Only Jira administrators can create projects with sample data.

Create a new project with sample data:

1. Log into Jira as the Jira administrator, **Dakota Jones**.
 - a. From the **Projects** menu, select **Create project**.
 - b. Select the **Scrum software development** template.
 - c. At the bottom of the Create project page, click **Create sample data**.
 - d. Select the **Scrum software development** template.
 - e. Click **Next**.
 - i. Name the project **Sample Scrum Project**.
 - ii. Leave the Key as **SSP**.
 - iii. Change the **Project Lead** to Max Taylor.
 - iv. Click **Submit**.
 - f. After a short wait, you should see your new project's backlog which is full of issues.
 - g. Add 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 **Users and roles**.
 - iii. On the **Users and roles** page, click **Add users to a role**.
 - iv. In the Users or groups field, start typing **max** and select **Max Taylor**.
 - v. In the role field, select **Administrators**.
 - vi. Click **Add**.

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.

1. Log in as the project administrator, **Max Taylor**.
 - a. In the main menu, click **Dashboards** and select **Manage Dashboards**.
 - b. In the **Manage Dashboards** page, click **Create new dashboard** in the top right of the page.
 - i. Name: **Development dashboard**.
 - ii. Click the **Add 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**.
 - v. Click **Add** to add all users who can access the Sample Scrum Project as viewers of this board.
 - vi. Click the **Add Editors** drop-down and select **User**.

- vii. Click the **Users** and add **Cassie Owens**.
- viii. Click **Add** to add Cassie as a Dashboard editor.
- ix. Click **Add** to create the Dashboard.

Add the Sprint Health gadget to the dashboard:

1. From the **Dashboards** dropdown menu, select Development dashboard.
2. From the **Development dashboard**, click **Add gadget**.
 - a. Click **Load all gadgets**.
 - b. Scroll down and view all the gadgets that are available to you.
 - c. Find the **Sprint Health Gadget** and click **Add gadget**.
 - d. Close the **Add a gadget** dialog by clicking outside of the **Add a gadget** dialogue, or clicking the X in the top right of the pop up.
 - e. 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 instance.
 - iv. Click **Save**.
 - f. Now you see the gadget displayed on your dashboard.

Add Assigned to me gadget.

1. In the right column, click **add a new gadget**.
 - a. Find the **Assigned to Me** gadget and click **Add gadget**.
 - b. Close the **Add a gadget** dialog by clicking outside of the **Add a gadget** dialogue or the X in the top right of the pop up.
 - c. Configure the **Assigned to Me** gadget:
 - i. Drag the **Priority** field to the top of the Columns to display list.
 - ii. Click the drop-down and type in **Fix versions**, then click to select it.
 - iii. Click **Save**.
 - d. In the Assigned to Me gadget, click **...** (three dots) and select **Edit**.
 - i. You decide to remove the Fix versions field so click the trash icon next to it.
 - ii. Click **Save**.

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.

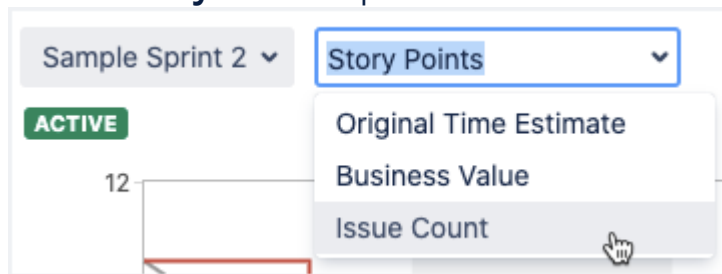
1. On your dashboard, click **Edit layout**.
 - a. Experiment with choosing different column layouts to see how the display changes.
 - b. Return to the default equal-sized two-column layout.

2. 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 **two diagonal arrows** in the gadget header.
 - ii. To restore the gadget to its original size, click the **two diagonal arrows** in the gadget header.
 - iii. To hide the results, click the **three dot** icon in the gadget header and select **Minimize**.
 - 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

Run the Burndown Chart report:

1. Go to the **Sample Scrum Project**.
 - a. 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.
 - b. In the project sidebar, click **Reports**.
 - c. 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 value it matches in the vertical y-axis.
 - d. Click the **Story Points** drop-down and select **Issue Count**.



- **Note:** Now you can see the work done and the remaining number of issues.
- e. Ensure **Show Non-Working Days** is unchecked (at the top right of the chart).
 - **Note:** Now you can see the work done and the remaining number of issues.
 - 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.
 - f. Click the sprint drop-down and select **Sample Sprint 1**.
 - **Note:** Here you can see how the team performed in sprint 1.
 - g. Scroll down and explore the details in the chart at the bottom.

Run the Velocity Chart report:

1. Click the **Switch report dropdown menu** (it's next to the Burndown Chart title). Select **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.
2. Return to the SSP board's **Active sprints** page.
 - a. Move all the issues and subtasks to the **DONE** column. If prompted, click **Update** to update the parent issue.
 - b. Click **Complete sprint**, then **Complete**.
 - **Note:** You are 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
 - c. Click the **Switch report** dropdown menu. Select **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.
 - d. If you scroll down to the bottom, you can click each sprint to go to the sprint report.

Run the Control Chart:

1. Click the **Switch report** dropdown menu. Select **Control Chart**.
 - a. 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 its 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).
 - b. Click one of the hollow dots on the graph to view a single issue's details.
 - c. Click one of the filled dots on the graph to view data on a cluster of issues.
 - d. Drag the cursor along a portion of the chart to zoom in on that area of the chart.

- e. 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 dashboards	https://confluence.atlassian.com/jirasoftwareserver/configuring-dashboards-939939002.html
View and understand the burndown chart	https://confluence.atlassian.com/jirasoftwareserver/release-burndown-938845669.html#ReleaseBurndown-ViewingtheReleaseBurndownreport
View and understand the velocity chart	https://confluence.atlassian.com/jirasoftwareserver/velocity-chart-938845700.html
View and understand the control chart	https://confluence.atlassian.com/jirasoftwareserver/control-chart-938845628.html

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, and how the team is performing overall. I have to go to different pages in the project to find out this information.	Create dashboards so you have one place to give you an overview of the data 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 8 Other Jira Features

- ① 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

If you completed the first lab in this course, you can use the existing Megastorm project. If you didn't create this project, use the instructions in [Lab 2 Exercise 1](#) to create the Megastorm project.

Exercise 1 Creating a Project with Shared Settings

i 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. Log in to Jira 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. From the Jira menu click **Projects > Create project**.
 - b. On the **Create project** screen, select the **Scrum software development** template.
 - c. At the bottom of the **Create project** screen click **Create with shared configuration**.

The screenshot shows the 'Create project' interface. It features a grid of project templates. Under the 'Software' category, 'Scrum software development' is selected. Below this, there are 'Service' and 'Business' categories with their respective templates. At the bottom of the screen, a row of buttons includes 'Import a project', 'Create with shared configuration' (highlighted with a red box), and 'Create sample data'. To the right of these are 'Next' and 'Cancel' buttons.

- d. In the Choose a project dropdown, select **Megastorm**.
- e. Click **Next**.
- f. Name the project **Space Music App**.

- g. Leave the key at **SMA**.
 - h. Leave the Project Lead as **Dakota Jones**.
 - i. Click Submit.
3. View project configuration:
 - a. 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

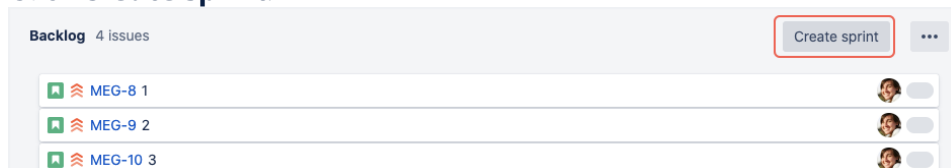
- ① You have two teams working from the same backlog. By enabling parallel sprints, each team can work on their own active sprint simultaneously.

Enable parallel sprints:

1. As the Jira administrator, **Dakota Jones**, click the cog icon (Settings) in the main menu and select **Applications**.
 - a. In the Applications sidebar, click **Jira Software configuration**.
 - b. If it hasn't already been enabled, check **Parallel Sprints**.

Run a parallel sprint:

1. Log in to Jira as the project administrator, **Max Taylor**.
 - a. Go to the **Megastorm** project.
 - b. In the project sidebar, click **Backlog**. Note that sprint 1 is in progress.
 - c. If your backlog is empty, add three issues using the Task or Story issue types.
 - **Note:** Recall that in [Lab 4](#) you updated the board's filter to remove Bugs from the MEG board. If you create a Bug, it won't appear on the board.
 - d. Click **Create sprint**.



2. In the **Create sprint** dialog, leave all fields at their default settings and click **Create**.
 - a. Drag some of the issues from the backlog into **MEG Sprint 2**.
 - b. Click **Start sprint**.
 - i. Accept the defaults and click **Start**.
 - c. Back on the **Active sprints** page, you now see all the issues in both active sprints.
 - d. Click **Switch sprints** 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
Starting a new project	https://confluence.atlassian.com/jirasoftwareserver/starting-a-new-project-938845217.html
Use parallel sprints	https://confluence.atlassian.com/jirasoftwareserver/using-parallel-sprints-938845513.html

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.