# JIRA – an Overview

Atlassian JIRA is a proprietary issue tracking system. It is used for tracking bugs, managing issues, and provides project management functionalities. There are many such tools available in the market, but the best thing about JIRA is that, it can be configured easily and it offers a wide range of customizations. Out of the box, JIRA offers defect/bug tracking functionality, but it can be customized to act like a helpdesk system, simple test management suite, or a project management system with end to end traceability.

*(For more resources related to this topic, see*[*here*](https://www.packtpub.com/books/content/jira-%E2%80%93-overview#more)*.)*

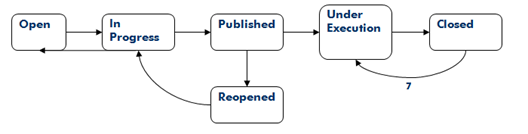
Applications, uses, and examples

The ability to customize JIRA is what makes it popular among various companies who use it. There are various applications of JIRA as mentioned as follows:

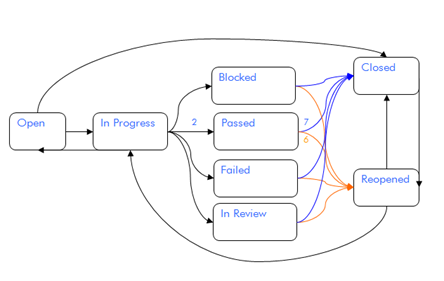
* Defect/bug tracking
* Change requests
* Helpdesk/support tickets
* Project management
* Test case management
* Requirements management
* Process management

Let's take a look implementation of test case management:

1. Issue Types:
   * **Test Campaign**: This will be the standard issue type
   * **Test Case**: This will be sub-task
2. Workflow for Test Campaign:



1. New states:
   * Published
   * Under Execution
2. Condition:
   * A Test Campaign will only pass when all the Test Cases are passed
   * Only reporter can move this Test Campaign to the Closed state
3. Post Function:
   * When the Test Campaign is closed send an email to everyone in a particular group
4. Workflow for Test Case:



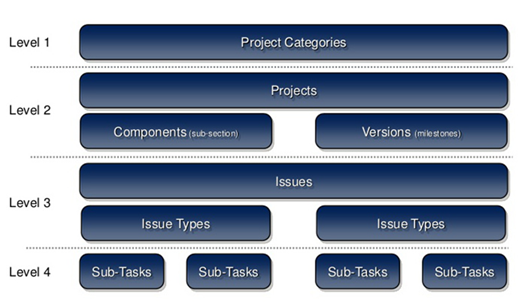
1. New States:
   * Blocked
   * Passed
   * Failed
   * In Review
2. Condition:
   * On the assign can move the Test Case to Passed state
3. Post Function:
   * When the Test Case is moved to Failed state, change the issue priority to Major
4. Custom Fields:

The following table shows the various fields and their descriptions:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Values** | **Field configuration** |
| Category | Select List |  |  |
| Customer Name | Select List |  |  |
| Steps to Reproduce | Text Area |  | Mandatory |
| Expected input | Text Area |  | Mandatory |
| Expected output | Text Area |  | Mandatory |
| Pre-Condition | Text Area |  |  |
| Post-Condition | Text Area |  |  |
| Campaign Type | Select List | Unit  Functional  Endurance  Benchmark  Robustness  Security  Backward Compatibility  Certification with Baseline |  |
| Automation Status | Select List | Automatic  Manual  Partially Automatic |  |

JIRA core concepts

Let us take a look at the architecture of JIRA in the following diagram; it will help you to understand the core concepts.



* **Project Categories**: When there are too many projects in JIRA, it becomes important to segregate them into various categories. JIRA will let you create several categories that could represent the business units, clients, or teams in your company.
* **Projects**: A JIRA project is a collection of issues. Your team could use a JIRA project to coordinate the development of a product, track a project, manage a help desk, and more, depending on your requirements.
* **Components**: Components are sub-sections of a project. They are used to group issues within a project into smaller parts.
* **Versions**: Versions are points-in-time for a project. They help you schedule and organize your releases.
* **Issue Types**: JIRA will let you create more than one issue types that are different from each other in terms of what kind of information they store. JIRA comes with default Issue Types like Bug, Task, and Sub-Task but you can create more Issue Types that can follow their own workflow as well as have different set of fields.
* **Sub-Tasks**: Issue Types are of two types—standard and sub-tasks, which are children of a standard task. For instance, you could have Test Campaign as standard issue type and Test Cases as sub-tasks.

Introduction to JQL

**JIRA Query Language**, better known as **JQL** is one of the best features in JIRA that lets you search the issues efficiently and offers lots of handy features. The best part about JQL is that it is very easy to learn, thanks to the autocomplete functionality in the Advanced Search that helps the user with suggestions based on keywords typed.

JQL consists of questions whether single or multiple , can be combined together to form complex questions.

Basic JQL syntax

JQL has a field followed by an operator. For instance, to retrieve all the issues of the project "CSTA", you can use a simple query like this:

*project = CSTA*

Now within this project, if you want to find the issues assigned to a specific user, use the following query:

project = CSTA and assignee = ravisagar

There might be several hundred issues assigned to a user and maybe we just want to focus on issues whose priority is either *Critical* or *Blocker*.

project = CSTA and assignee = ravisagar and priority in (Blocker, Critical)

What if instead of issues assigned to a specific user we want to find the issues assigned to all other users except one? It can be achieved in the following way:

project = CSTA and assignee != ravisagar and priority in (Blocker, Critical)

So you can see that JQL consists of one or more queries.

Project reports

Once you start using JIRA for issue tracking of any type, it becomes imperative to derive useful information out of it. JIRA comes with built in reports that show real time statistics for projects, users, and other fields.

Let's take a look at each of these reports.

Open any project in JIRA that contains lot of issues and has around 5 to 10 users, which are either assignee or reporters. When you open any project page, the default view is the **Summary** view that contains a 30 day Summary report and **Activity Stream** that shows whatever is happening in the project like creation of new issues, update of status, comments, and basically any change in the project.

On the left hand side of the project summary page, there are links for **Issues** and **Reports**.

Average Age Report

This report displays the average number of days for which issues are in unresolved state on a given date.

Created vs. Resolved Issues Report

This report displays the number of issues that were created over the period of time versus the number of issues that were resolved in that period.

Pie Chart Report

This chart shows the breakup of data. For instance, in your project, if you are interested in finding out the issue count for all the issue types, then this report can be used to fetch this information.

Recently Created Issues Report

This report displays the statistical information on number of issues created so for the selected period and days. The report also displays status of the issues.

Resolution Time Report

There are cases when you are interested in understanding the speed of your team every month. How soon can your team resolve the issues? This report displays the average resolution time of the issues in a given month.

Single Level Group By Report

It is a simple report that just lists the issues grouped by a particular field like **Assignee**, **Issue Type**, **Resolution**, **Status**, **Priority**, and so on.

Time Since Issues Report

This report is useful in finding out how many issues were created in a specific quarter over the past one year and, not only that, there are various date based fields supported by this report.

Time Tracking Report

This comprehensive report displays the estimated effort and remaining effort of all the issues. Not only that, the report will also give you indication on the overall progress of the project.

User Workload Report

This report can tell us about the occupancy of the resources in all the projects. It really helps in distributing the tasks among **Users**.

Version Workload Report

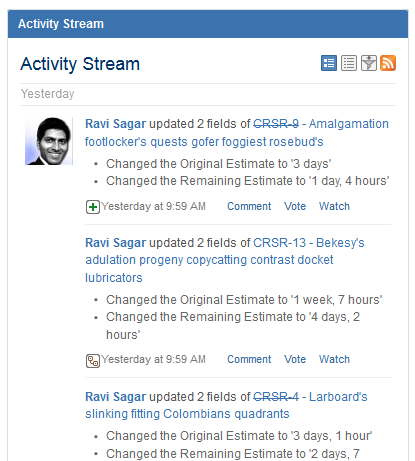
If your project has various versions that are related to the actual releases or fixes, then it becomes important to understand the status of all such issues.

Gadgets for reporting

JIRA comes with lot of useful gadgets that you can add in the dashboard and use for reporting purpose. Let's take a look at some of these gadgets.

Activity Stream

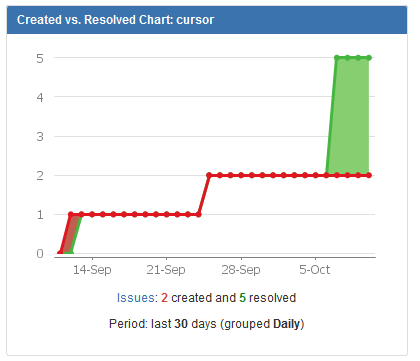
This gadget will display all the latest updates in your JIRA instance. It is also possible to limit this stream to a particular filter too. This gadget is quite useful, as it displays up to the date information in the dashboard.



Created vs. Resolved Chart

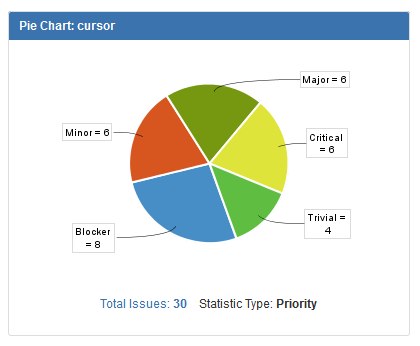
The project summary page has a chart to display the issues that were created and resolved in the past 30 days. There is a similar gadget to display this information.

You can also change the duration from 30 days to whatever you like. This gadget can be created for a specific project.



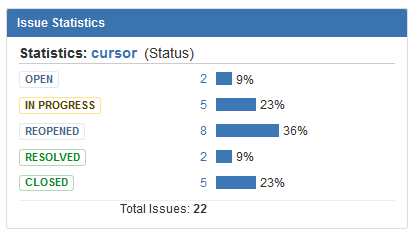
Pie Chart

Just like the Pie Chart, which is available in the project reports, there is a similar gadget that you can add in the dashboard. For instance, for a particular project, you can generate the Pie Chart based on **Priority**:



Issue Statistics

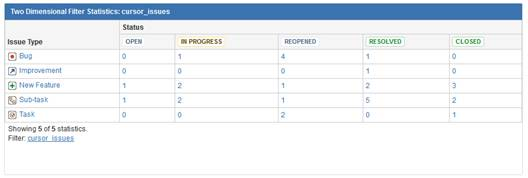
This gadget is quite useful in generating simple statistics for various fields. Here, we are interested in finding out the breakup of project in terms of issue status:



Two Dimensional Filter Statistics

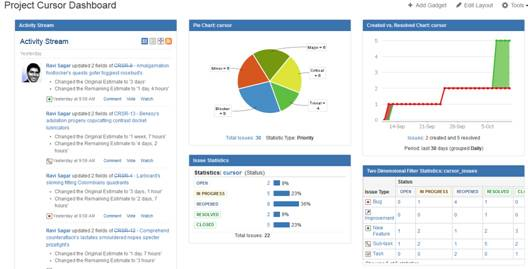
The Issue Statistics gadget can display break up of project issues for every Status. What if you want to further segregate that information? For instance, how many issues are open and to which **Issue Type** they belong to? In such scenarios, **Two Dimensional Filter Statistics**can be used.

You just need to select two fields that will be used to generate this report, one for *x* axis and another for *y* axis.



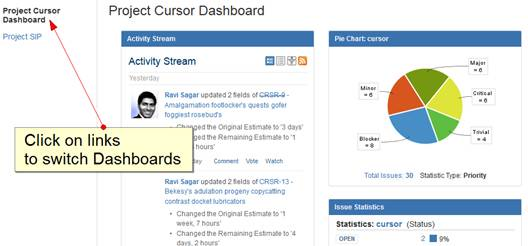
These are some of the common gadgets that can be used in the dashboard; however, there are many more gadgets. Click on the **Add Gadget** option on top-corner to see all such gadgets in your JIRA instance. Some gadgets come out of the box with JIRA and others are part of add-ons that you can install.

After you select all these gadgets in your dashboard, it will look like the following screenshot:



This is the new dashboard that we have just created and configured for a specific project, but it is also possible to create more than one dashboard. Just click on the **Create Dashboard**option under **Tools** on top right-hand side corner to add another dashboard.

If you have more than one dashboard, then you can switch between them using the links on the left-hand side of the screen as shown in the following screenshot:



Simple CSV import

Lets us understand how to perform a simple import of CSV data. The first thing to do is prepare the CSV file that can be imported in JIRA. For this exercise, we will import the Issues in a particular project; these issues will have data like **Issue Summary**, **Status**, **Dates**, and few other fields.

Preparing the CSV file

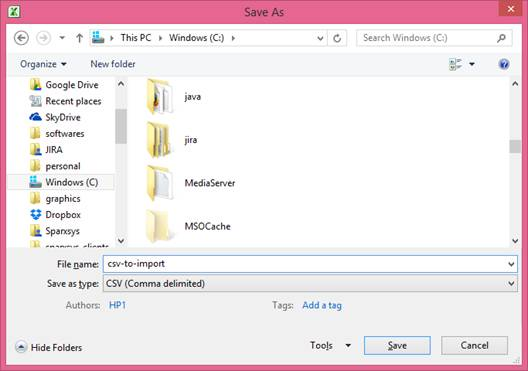
We are going to use MS Excel for preparing the CSV file with the following data:



If your existing tool has the option to export directly into the CSV file then you can skip this step, but we recommend reviewing your data before importing it into JIRA. Usually, CSV import will not work if the format of the CSV file and the data is not correct.

It is very easy to generate a CSV file from an Excel file. Follow these steps:

1. Go to **File** | **Save As** | **File name:** and select **Save as type:** as **CSV (comma delimited)**. If you don't have Microsoft Excel installed, you can use Libre Office Calc, which is an open source alternative for Microsoft Office Excel.



1. You can open the CSV file to verify its format too:

Mastering JIRA

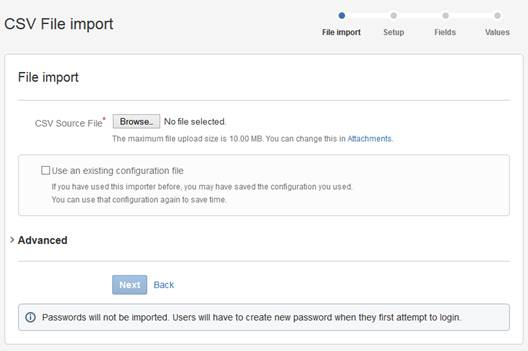
Our CSV file has the following fields:

|  |  |
| --- | --- |
| **CSV Field** | **Purpose** |
| Project | JIRA's project key needs to be specified in this field |
| Summary | The issue summary is mandatory and needs to be specified in the CSV file |
| Issue Type | It is important to specify the Issue Type |
| Status | This is the status of the Issue; these are the workflow states that need to exist in JIRA and the project workflow should have states that are going to be imported through a CSV file |
| Priority | The priorities mentioned here should exist in JIRA before import |
| Resolution | The resolutions mentioned here should exist in JIRA before import |
| Assignee | This specifies the Assignee of the Issue |
| Reporter | This specifies the Reporter of the Issue |
| Created | This is the Issue creation date |
| Resolved | This is the Issue resolution date |

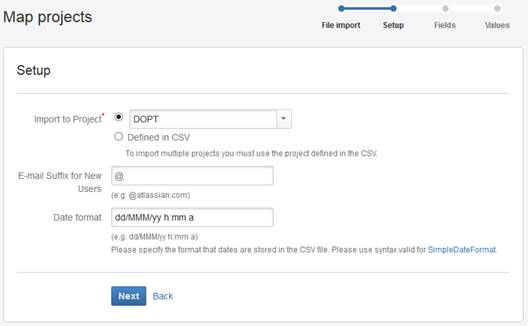
Performing CSV import

Once your CSV file is prepared, then you are ready to perform the import operation in JIRA:

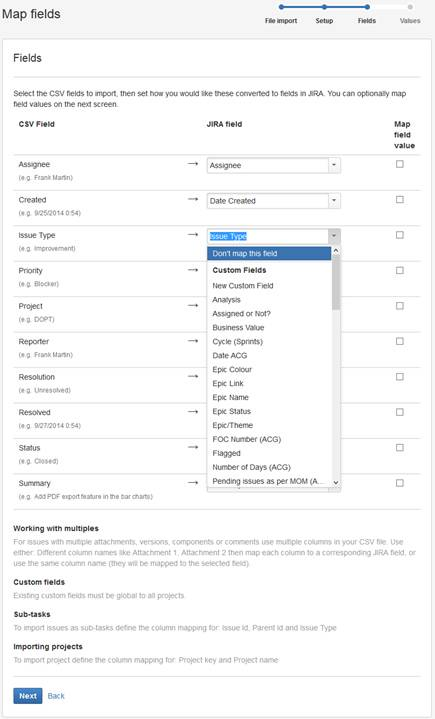
1. Go to **JIRA Administration** | **System** | **External System Import** | **Import from Comma-separated values (CSV)** (under **IMPORT & EXPORT**):
2. In the **File import** screen, for **CSV Source File** field, click on the **Browse** button to select the CSV file that you just prepared on your machine.
3. Once you select the CSV file, the **Next** button will be enabled:



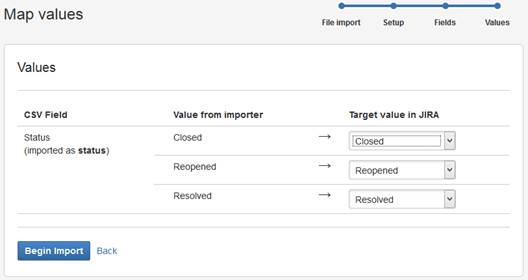
1. In the **Setup** screen, select **Import to Project** as **DOPT**, which is the name of our project.



1. Verify the **Date format** and it should match the format of the date values in the CSV file.
2. Press the **Next** button to continue.
3. In the **Map fields** screen, we need to map the fields in the CSV file to JIRA fields. This step is crucial because, in your old system the field name can be different from JIRA fields, so in this step map these fields to the respective JIRA fields.



1. Press the **Next** button to continue.
2. In the **Map values** screen, map the values of the **Status**, in fact this mapping of field values can be done for any field. In our case, the values in the status field are same as in JIRA so click on **Begin Import** button.



1. You will finally get the confirmation that issues have been imported successfully.

Mastering JIRA

If you encounter any errors during CSV import, then it is usually due to some problem with the CSV format. Read the error messages carefully and correct those issues. As mentioned earlier, the CSV import needs to be performed on test environment first.

User and group management

JIRA is a web based application to track project issues that are assigned to people. The users to whom these issues will be assigned need to exist in the system. When JIRA is deployed in any organization, the first thing that should be done is to gather the list of people who would be using this tool; hence, their accounts need to be created in JIRA. Each user will have their username and password that is unique to them and allows them to login to the system. JIRA has its own internal authentication mechanism as well as the ability to integrate with **Lightweight Directory Access Protocol** (**LDAP**).

Summary

In this article, we got introduced with JIRA, and various concepts of JIRA. We also got introduced with JQL, project reports and the tools used for these reports. Finally, we learned how to import a simple CSV file in JIRA.