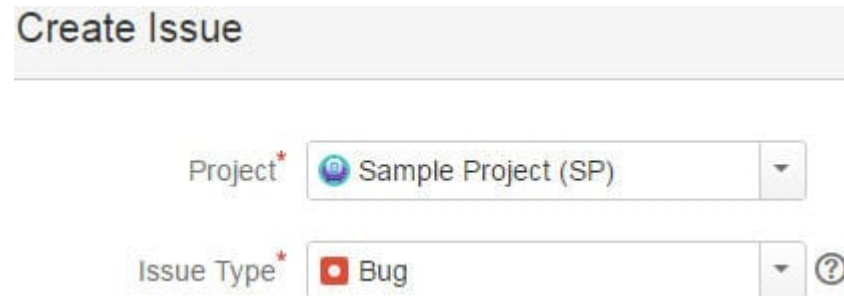


The Best Features of JIRA Bug Tracking Tool

Here we go.

#1) JIRA treats all work inside it as an Issue

So, in JIRA to create a defect would be to create an issue of the type “**Bug**”.



The screenshot shows the 'Create Issue' form in JIRA. The 'Project' field is a dropdown menu with 'Sample Project (SP)' selected. The 'Issue Type' field is a dropdown menu with 'Bug' selected. Both fields have a red asterisk indicating they are required. There is also a help icon (question mark) next to the 'Issue Type' dropdown.

#2) Defect reporting needs the following information recorded for every issue:

- Defect ID
- Defect title
- Defect description (steps to reproduce)
- Environment information
- Screenshot(attachment)
- Severity
- Assign it to someone
- **Status**- All the statuses in the bug life cycle

All the options are available to be able to create a defect effectively.

Please note the fields highlighted in Red below:

Create Issue Configure Fields

Summary

Component/s **None**

Description

Priority **Medium**

Environment

Attachment

Assignee **Automatic**

Create another Create Cancel

The two fields you are not seeing here are:

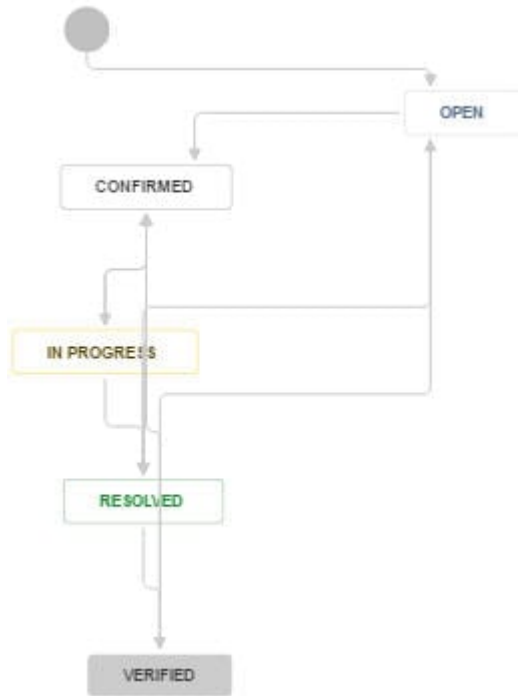
- Defect ID
- Status

These two fields are auto-created by JIRA. All issues will have a unique ID assigned to them by JIRA. Status of all issues is “To-Do” or “New” in JIRA by default on creating a bug.

Therefore, all the common facilities for defect reporting are available in JIRA too. In fact, more options such as labels, linking defects, estimating efforts can be used.

#3) Defect Life Cycle:

All bug life cycle statuses as in **Bugzilla** (or any other **popular bug tracker**) can be accomplished here too:



This will need a little bit of customizing by your JIRA admin, but it is easy to do. For those, do not want to bother with the customization, you can't go wrong with the default set up as well.

#4) Comments and collaboration with the Dev Team

Every issue, its updates, people assignment, comments received from the Dev team – everything is tracked in JIRA under the activity log.

This allows for better visibility and collaboration with the development teams:



#5) Linking the defect to a requirement to enable traceability

Link option in the JIRA issue fields lets you link a particular issue to another one. Let's say if Defect 2 is a duplicate of Defect 1 you can establish that relationship.

Similarly, if a defect is blocking a requirement or is related to a requirement – you can make this aspect visible in JIRA.

Link

JIRA Issue

Web Link

Select a JIRA issue to link this issue to

This issue

Issue

Comment

relates to

relates to

blocks

is blocked by

clones

is cloned by

duplicates

is duplicated by

causes

is caused by

Link Cancel

The resulting links will appear in the issue details page as below:

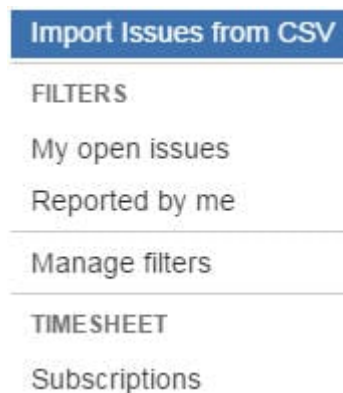
Issue Links				
is caused by	PROJ-5 Create a better HTML CSS	↑	TO DO	
is duplicated by	PROJ-4 FAQ page rendition fails on IE	↑	TO DO	
relates to	PROJ-1 FAQs section has be include in the main menu next to the che...	↑	TO DO	

The relationship types are self-explanatory and the usage of *simple-common-everyday-language* words (such as relates to, caused by, etc.) makes it super easy and intuitive for any JIRA user to use this right.

#6) Defects can be imported from a CSV file

This aids the bulk creation of issues in JIRA at once. Also, if your team is new and you don't want them creating issues directly into the tool, you can have them report the defects in an excel sheet. Once they are reviewed and confirmed as valid, they can be imported all at once into the tool using this functionality.

Whichever way you use it, this is a big plus.



#7) Defects can be exported into Word, XML, and printable formats



This supports better portability of your defect data, especially useful if you want to share your defect data with people who are non-JIRA users.

#8) Comprehensive Issue Reports:

In addition, if you need reports go to "*Project – reports*" and generate all sorts of reports as below:

Issue analysis



Average Age Report

Shows the average age of unresolved issues for a project or filter. This helps you see whether your backlog is being kept up to date.



Created vs. Resolved Issues Report

Maps created issues versus resolved issues over a period of time. This can help you understand whether your overall backlog is growing or shrinking.



Pie Chart Report

Shows a pie chart of issues for a project/filter grouped by a specified field. This helps you see the breakdown of a set of issues, at a glance.



Recently Created Issues Report

Shows the number of issues created over a period of time for a project/filter, and how many were resolved. This helps you understand if your team is keeping up with incoming work.



Resolution Time Report



Single Level Group By Report



Time Since Issues Report

If we have to review JIRA's analytics in one word, it's fantastic.

Advanced/Power users of JIRA can also create advanced search filters to generate deeper insights.

For example, if want to look at all the defects assigned to you across multiple projects (BM and AB), you could use a JQL query like below:

project IN (BM, AB) AND issuetype = Bug and assignee=currentuser() and priority in (blocker, critical) and status in ("To Do", open) ? Q Basic

1-9 of 9 Columns

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated
	BM-132	Test	Swati Seela [Administrator]	Swati Seela [Administrator]		OPEN	Unresolved	17/Feb/16	03/Mar/16
	AB-38	Rest_validatAccount_creditcard_ ; credit card validation gives incorrect error messgae when Expiry date is prior than the present date.	Swati Seela [Administrator]	Test		TO DO	Unresolved	13/Aug/15	19/Aug/15
	AB-37	Update functionality for SSN,accets input'0'with success,but does not update in DB	Swati Seela [Administrator]	Test		TO DO	Unresolved	13/Aug/15	19/Aug/15
	AB-35	Timeout Error for server socket	Swati Seela [Administrator]	Test		TO DO	Unresolved	13/Aug/15	19/Aug/15
	AB-29	Buy credit card accepts negative amount and it's adding in available limit in database	Swati Seela [Administrator]	Test		TO DO	Unresolved	13/Aug/15	19/Aug/15
	AB-27	In CC card restful service "Add account" Credit limit ,available limit and cash limit accepts 0 and show "msg": "Scuccessfully saved credit card information".	Swati Seela [Administrator]	Test		TO DO	Unresolved	13/Aug/15	19/Aug/15
	AB-26	In CC restful service "add account" CC expiry date should be greater than the current date but accepts the date prior to the current date.	Swati Seela [Administrator]	Test		TO DO	Unresolved	13/Aug/15	19/Aug/15
	AB-19	If we give non matching name and credit card number i.e name of one person and registered credit card of other person the amount in the request get deposited in the account of name provided.	Swati Seela [Administrator]	Test		TO DO	Unresolved	13/Aug/15	19/Aug/15

So all in all, bug tracking/defect management in JIRA is very similar if not superior to dedicated bug trackers. Next time you have to work on it, don't worry. You are in good hands.

Applicability of JIRA to Testing – An Alternative Dilemma

While this is one side of the coin, there is definitely another dimension to how people view the applicability of JIRA to QA or testing.

When you ask a group of QAs, "What is JIRA?"- Many will answer that JIRA is a defect tracking tool. Mind you, I have heard this from many senior QA professionals. This might be from the fact that, defect management/tracking is all they might have used JIRA for.

But, there is a lot more to it. When used right, core JIRA with its agile capabilities can be your one-stop-shop for high-level project management.

It can really support requirement tracking and progress, bug tracking, estimating, sprint tracking through SCRUM & KANBAN boards, reporting and collaborating.

You might be using a tool for one thing, but next time try and learn a few things around and about the tool that will help you understand and use it better.

So, as a next step, you could explore a few other cool features of JIRA (that might not directly be related to bug tracking) that might make it your go-to choice.

- Customizable Dashboards
- Test Management Add-ons
- Vote and Watching an issue
- Time tracking
- Agile Project and Scrum boards
- Confluence/Documentation support integration, etc.

Creating a Jira Issue and Various Fields

Jira Issues: Different Types of Jira Issues

Jira gives you very simple ways to create/log issues.

It not just allows us to file bugs but also enables us in other kinds of 'tickets' or 'requests'. It is more of a general request management application.

This tutorial will explain more on Issue types in Jira, creating an issue, different fields on 'Create Issue' page and their details in simple terms with pictorial representation for your easy understanding.



© www.SoftwareTestingHelp.com

Jira Issues

Different organizations may have different types of issues depending upon their suitability/ needs. A Jira administrator can efficiently customize this field.

Issues can be of different types and given below are the Description/meaning of Issue types:

1. **Bug:** This is any defect or deviation that is found in the application.
2. **Enhancement request:** It is also known as a change request (CR). This type is used to depict any change in the existing functionality or altogether a new functionality.
3. **Task:** This is more of a configuration or analysis issue. **For Example**, setting up proper configurations can be a task.
4. **Question:** Issue can be as simple as asking a question about how to use some functionality in the application. This type is more often used by the end customers.
5. **Epic:** This is normally a huge issue which is ideally broken down into several small issues. It can take several sprints to complete the main epic issue in an agile environment.
6. **Financial Object:** Often project/product management uses this type of issue to track their finances.
7. **Story:** Entire user story about a feature could be a type of issue.
8. **Test case:** Issue can be a test case. This type of issue will be available once Jira is integrated with plug-ins like Zypher.

Creating an Issue

Assuming that a user has logged into Jira and the desired project.

Step 1:

Click on '+' ('Create') toolbar button.

This will display a screen/page as shown in the below image:

The screenshot shows the 'Create issue' page in Jira. On the left is a blue sidebar with 'Jira Software' and navigation links: Dashboards, Projects, Issues, and Settings. The main content area is titled 'Create issue'. It contains two required dropdown menus. The first is 'Project*' with a red asterisk and a red exclamation mark icon; it shows 'TestProject (TESTP...' with a callout box saying 'Project name needs to be chosen from the drop-down'. The second is 'Issue Type*' with a red asterisk and a red exclamation mark icon; it shows 'Bug' with a callout box saying 'Issue type needs to be selected from the drop-down'. At the bottom of the form are 'Next' and 'Cancel' buttons.

On this page, select project and issue/request type and then click on the 'Next' button.

This will open up the 'Create issue' page as displayed in the following images:

Create issue

Project name

Project
TestProject





Issue Type
Bug

Summary*


Issue title

Component/s
None

Description

Style^v | B | I | U | A^v | A^v |  |  |  |  | +^v

I

 ?

Fix Version/s
None






Priority*


↑ Medium ▼ ?

Labels

Begin typing to find and create labels or press down to select a suggested label.


Environment

Style **B** *I* U **A** [~] **A** [~]     

 ?

For example operating system, software platform and/or hardware specifications (include as appropriate for the issue).

Attachment

 Drop files to attach, or [browse](#).


Affects Version/s

None

Linked Issues


blocks ▼

Issue

▼ + 

Begin typing to search for issues to link. If you leave it blank, no link will be made.

Assignee

 Automatic ▼ [Assign to me](#)

Epic Link

▼

Choose an epic to assign this issue to.

Sprint

▼

Jira Software sprint field

Team

...

The Portfolio for Jira team that will work on this issue.

Estimate at start

▼

How much effort/time required for the issue to fix - starting estimate

Step 2:

Enter the mandatory details and other data as much as possible on the 'Create issue' page.

Step 3:

Click on the 'Create' button. This will generate a unique issue ID. ID will consist of project identifier concatenated with numeric digits.

In the above Example, the project chosen is 'TestProject', hence the ID could be like 'TESTPROJ1234'.

- Once the issue is created, thereafter it can be searched using the issue ID.

Description of Fields on 'Create issue' Page

(Create issue page images are divided into 3 parts for better readability).

Note: Jira administrator and/or developer can add/remove the custom fields depending upon the organization needs.

#1) Summary:

This is also more often called as the title of the issue and is a very important field of a Jira issue.

The title should be as unique and precise as possible so that by looking at the title itself, the issue can be understood. This helps the bug review board and/or product owners to prioritize and assign the issue without looking deep into it.

#2) Component/s:

Name(s) of the module or area of the application where the defect is detected in case of the 'Bug' issue type.

It could be the area where the changes are required in case of a CR. This is usually a drop-down consisting of different modules/components which exist in the application. Project person has to get it populated from the administrator.

#3) Description:

Typically should contain the steps to reproduce the issue if the issue type is a bug.

In case of an enhancement request, it should detail about the new requirement which is typically called as a story in the agile terminology. Ideally, this field should be updated regularly during the course of the issue workflow.

#4) Fix Versions:

Name of the version in which the issue/enhancement request will be delivered. This value is typically filled by the product owner in co-ordination with the scrum master in an agile scrum environment.

#5) Priority:

This field indicates the criticality of the issue.

It can be a show stopper, meaning application testing cannot go ahead in a testing phase. The crash of an application is an ideal **Example** of a 'Show Stopper' (critical) issue.

Bug review board and the product owners have every right to change the priority of the issue. This field is a drop-down list with values like 'Low', 'Medium' ('Major'), 'Critical', 'Trivial' etc.

#6) Labels:

This field is entered with the texts which will help in categorizing issues.

#7) Environment:

This is an optional field and the test environment is specified here.

#8) Attachment:

Supporting images for the issue being created. The user can simply drag and drop images or copy and paste.

#9) Affects Version/s:

For a 'bug' type of issue, the product version should be entered here.

For Example 5.6, 5.7 etc.

#10) Linked Issues:

Other relevant issues can be linked to the new issue by choosing a proper value from this drop-down.

For instance, if the issue is introduced by a fix of some other issue then the value to be chosen from the drop-down could be 'Introduced By'. This field becomes extremely important if a new defect is triggered by some fix or enhancement.

=> Issue: After selecting a proper value in 'Linked issues', relevant issue ID is mentioned here.

#11) Assignee:

It is the name of the user who will be working on the issue.

For instance, in the case of a bug, it will be the name of the developer who will fix the issue. This field is typically filled by the product owner or scrum master. Again who assigns the issue may vary from one organization to another.

=> Clicking on 'Assign to me' (located at the right corner of 'Assignee' field) will assign the issue to the logged in user.

#12) Epic Link:

Choose the relevant link of the epic.

#13) Sprint:

Name of the sprint is selected here, indicating when the issue will be worked upon. It could be a future sprint as decided by the product owner.

#14) Team:

There can be different teams, in an Agile environment. The issue is assigned to one of the teams. This assignment is usually done by the product owner or scrum master in coordination with the product owner.

#15) Estimate at the Start:

This field will indicate how much time effort will be required to resolve the issue.

More often called as 'guesstimate'. This will consist of the required testing efforts also. It could be mentioned in hours/days/weeks or story points. In an agile environment during sprint planning, the entire team reaches a common guess.

#16) Reporter:

This field is auto-populated by Jira with the name of the logged in user.

Note: We could have some other custom fields as below (which are not seen in the above images):

(i) Environment type:

Indicates if a defect is found in a test or production environment.

This field values may vary from organization to organization. If Jira is used to create issues only internally in the organization and not by end customers, then this field may not exist at all.