

Xray is a **test management tool used inside Jira**.

In simple words, Xray helps teams **manage test cases, test execution, and test results directly in Jira**, instead of using separate testing tools.

Xray allows you to create and manage test cases in Jira.

It helps you plan testing for a sprint or release.

It lets you execute tests and record results like pass or fail.

It gives clear testing reports and traceability.

The screenshot shows the Xray configuration interface. It starts with a 'Quick Setup' section with a 'Configure Project' button. Below it is a 'Follow these 3 steps to configure Xray' section:

- From project settings, go to Xray settings.** This step shows a screenshot of the Jira sidebar with the 'Xray Settings' option highlighted. A green arrow points from the sidebar to the 'Xray Settings' link in the main text.
- Add Xray Issue Types to your project.** This step shows a screenshot of the 'Xray Settings' page under the 'Summary' tab. It displays a list of available issue types: Test, Precondition, Test Set, Test Plan, Test Execution, and Sub Test Execution. Descriptions for each type are provided.
- Set Coverable Issue Types.** This step shows a screenshot of the 'Test Coverage' page. It lists available issue types (Test, Precondition, Test Set, Test Plan, Test Execution, Sub Test Execution) and selected coverable issue types (Test, Test Set, Test Plan, Test Execution, Sub Test Execution). A note at the bottom says: "A testable issue type is an issue that you can cover with tests in order to validate its associated criteria. To add a testable issue type, click the 'Coverable issue type' checkbox next to the issue type name." There is also a 'Next' button at the bottom.

*Configurations may differ if you use a [Company-Managed project](#) or a [Team-Managed project](#).

Try the Demonstration Project

Explore a fully pre-populated project environment to quickly get you started with Xray.

[Create a Project](#)

Join a live Demo

Effortlessly learn how Xray works end-to-end in a complete Xray Product Walkthrough.

[Sign up](#)

Configure xray in jira:

Install Xray

Enable issue types

Configure screens

Create Tests

Execute Tests

Track reports

Once configured, Xray becomes a **complete testing solution inside Jira**.

Xray Settings

Summary

Miscellaneous

Remote Jobs Trigger

Test Coverage

Defect Mapping

Test Types

Test Statuses

Test Step Statuses

Document Generator

Test Environments

Test Step Fields

Test Run Custom Fields

Parameter value lists

BDD Step Library

Summary

This page contains a brief summary of the Xray configurations for this project.

Issue Types

Xray Issue Types in Project

There are 6 Xray issue types configured for this project. Click [here](#) to edit the issue types for this project.

Name	Description	Present in Project
Test	This is the Xray Test Issue Type. Used to define test cases of different types that can be executed multiple times using Test Execution issues.	
Precondition	This is the Xray Precondition Issue Type. Used to abstract common actions that must be ensured before the test case execution. A Precondition can be associated with multiple test cases.	
Test Set	This is the Xray Test Set Issue Type. Creates a group of test cases. Used to associate all included Tests with other Xray issue types like Test Execution and Test Plan. A Test Set can also be associated with a requirement issue to provide coverage and test status.	
Test Plan	This is the Xray Test Plan Issue Type. Used to define the scope of test cases for a given test campaign and to aggregate all executions for those tests displaying the latest result for each test case.	
Test Execution	This is the Xray Test Execution Issue Type. Used to execute test cases already defined.	
Sub Test Execution	This is the Xray Sub Test Execution Issue Type. Used to execute test cases already defined. A Sub Test Execution can be created for a parent issue like a requirement in order to execute the test cases associated with it.	

The main issue types of Xray are:

Test

Used to create a test case. It contains test steps, test data, and expected results.

Test Execution

Used to run test cases. It stores the execution result like pass, fail, or blocked.

Test Plan

Used to group and manage test cases for a release or milestone.

Precondition

Used to define conditions that must be true before running a test, such as “user must be logged in”.

Create Test Execution

Required fields are marked with an asterisk *

Space *

xray-sample (XS)

Work type *

Test Execution

- Task
- Story
- Bug
- Epic
- Test
- Test Set
- Test Plan
- Test Execution
- Precondition

Your work type hierarchy determines the work items you can select here.

Create another

Cancel Create

Creation test case in xray :

Create Test issue

Add test steps

Add expected results

Link to story

(Optional) Add precondition

That's how a test case is created using Xray in Jira.

Create Test

Required fields are marked with an asterisk *

Space *
xray-sample (XS)

Work type *
Test

Status
Backlog

This is the initial status upon creation

Summary *
check login activity

Parent
Select parent

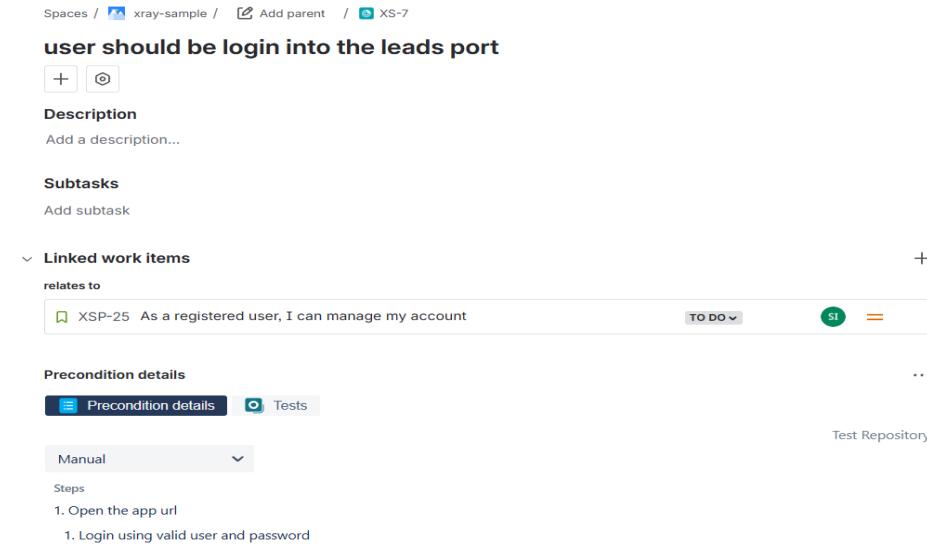
Your work type hierarchy determines the work items you can select here.

Components
 Create another

Cancel **Create**

Precondition in xray

A Precondition is like defining step 0 for your Tests. This is very useful in many use cases where you have to start by doing the same thing.



Spaces / xray-sample / Add parent / XS-7

user should be login into the leads port

+

Description
Add a description...

Subtasks
Add subtask

Linked work items +
relates to
XSP-25 As a registered user, I can manage my account **TO DO** SI =

Precondition details ...

Precondition details **Tests**

Test Repository

Manual

Steps

1. Open the app url
1. Login using valid user and password

Test Repository in xray

Go to your Jira project

Click Xray → Test Repository

Create folders

Drag and drop Test issues into folders

The screenshot shows the Xray Test Repository interface. On the left, there's a sidebar with 'FOLDERS' and 'TEST SETS' buttons. The main area shows a tree view of a 'Test Repository' folder containing 'Login', 'Oauth2', and a selected 'manual' folder which has 1 issue. The issue details show a test case titled 'XS-2 verify the leads are created successfully' with status 'MANUAL'. Below the tree view are buttons for '+ Create Test', '+ Create Precondition', and '+ Test Cases [BETA]'. At the top right, there's a 'Test Cases' button.

Test Repository only organizes Test issues.

It does not execute tests — execution happens in Test Execution.

Test Sets in xray :

A Test Set is an issue type in Xray that holds multiple Test cases.

It helps you quickly select and reuse the same group of tests again and again.

The screenshot shows the Xray Test Set interface for 'TestSet1-check'. The left panel displays the test set's details: 'Description' (Desc-Testset1), 'Subtasks' (Add subtask), and 'Linked work items' (a list of three Jira issues: XSP-21, XSP-25, and XSP-15). The right panel shows the 'Details' section with fields: Assignee (Sai Ganesh Immidisetti), Reporter (Sai Ganesh Immidisetti), Development (Open with VS Code, Create branch, Create commit), Labels (None), Due date (None), Start date (None), and Priority (Low). There are also 'Backlog' and 'More fields' sections.

Test Set in Xray = reusable group of test cases.

Test Plan in xray :

A Test Plan is an Xray issue type that groups test cases planned for a version or sprint.
It helps track which tests are in scope and their execution status.

Test Plan in Xray helps teams **organize, track, and control testing for a release inside Jira**.

The screenshot shows the Xray Test Plan interface. At the top, there are navigation buttons: Back, Add parent, XS-10, and a search bar. Below the header, the title "sprint-1-test-plan" is displayed. On the left side, there are sections for Description (with a description of "sprint-1-test-plan-desc" and status "in scope"), Subtasks (with a link to "Add subtask"), and Linked work items (listing three items: XS-2, XSP-47, and XS-6, each with a status dropdown and a priority indicator). On the right side, the "Details" section is expanded, showing fields for Assignee (Unassigned, Assign to me), Reporter (Sai Ganesh Immidisetti, Open with VS Code, Create branch, Create commit), Development (None), Labels (None), Due date (None), Start date (None), Priority (Medium), and More fields (Original estimate: 0m, Time tracking: No time loaded). A vertical sidebar on the right indicates the current tab is "Backlog".

Parameters in xray :

Test parameters allow you to define variables in a test, such as username, password, browser, or country.

During execution, you provide different values for these parameters.

Enter username = \${username}

Enter password = \${password}

Execution values:

admin / admin123

user / user123

Dataset for Test XS-2

The screenshot shows the Xray Dataset editor for Test XS-2. At the top, there are buttons for Create parameter, Import, and a three-dot menu. Below the header, the dataset table has columns: #, username, ..., password, ..., status, and There is one row with ID 1, where all fields are set to "None". At the bottom left, there is a "New" button with a plus sign.

#	username	...	password	...	status	...
1	None		None		None	

Info There are a total of 2 iterations to execute.

#	username	password	status	...
1	admin123	admin	vaid	
2	user123	adm	notvalid	

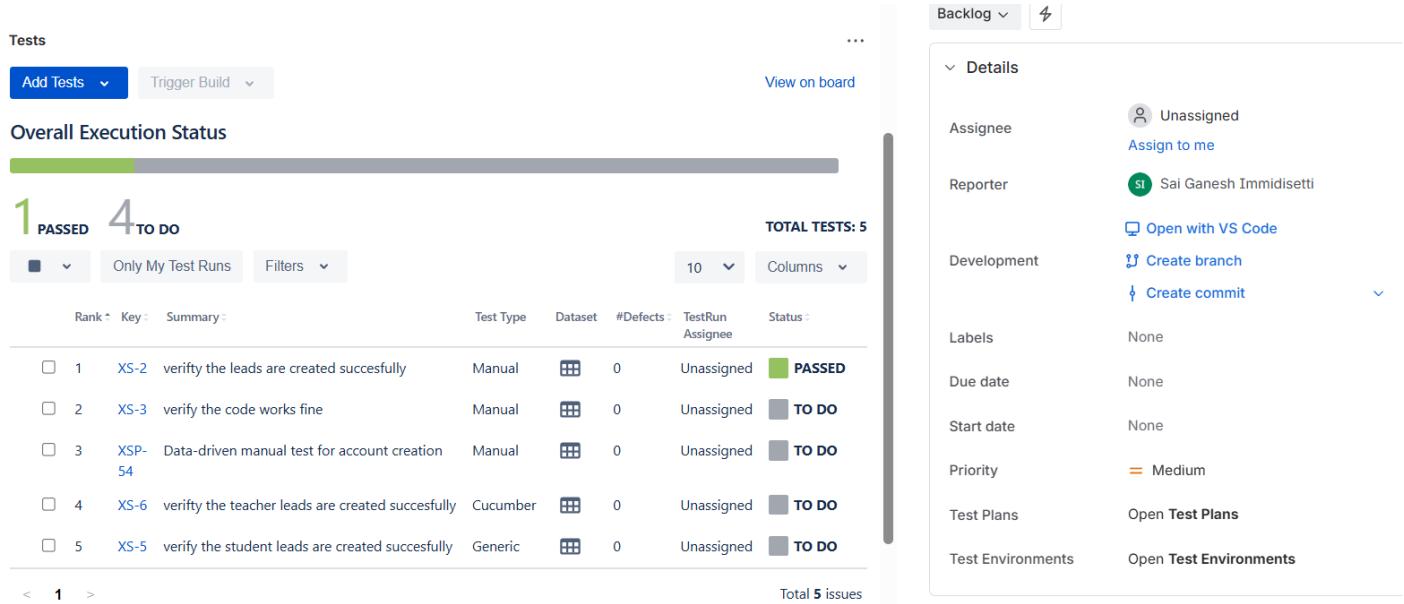
New 

Test Execution in xray :

A Test Execution is an Xray issue type that contains the tests you want to run. It stores execution results like pass, fail, or blocked.

Test Execution in Xray is used to **run test cases and record their results**.

In simple words, Test Execution is where **actual testing happens**.



The screenshot shows the Xray interface for managing test executions. On the left, there's a list of test cases with columns for Rank, Key, Summary, Test Type, Dataset, #Defects, TestRun Assignee, and Status. The status column uses color-coded icons: green for PASSED, grey for TO DO, and grey with a red exclamation mark for DEFECT. A summary bar at the top indicates 1 PASSED and 4 TO DO. On the right, there's a detailed view panel for a selected test case, showing fields for Assignee (Unassigned), Reporter (Sai Ganesh Immidisetti), Development (with options for Open with VS Code, Create branch, Create commit), Labels (None), Due date (None), Start date (None), Priority (Medium), Test Plans (Open Test Plans), and Test Environments (Open Test Environments).

Create a Test Execution issue

Add Test cases to it

Run each test step by step

Mark the result

Save evidence and comments if needed

All work items

	Work	Assignee	Reporter	Priority
<input type="checkbox"/>	XS-13 sprint-1 test execution	 Unassigned	 Sai Ganesh Immi...	 Medium
<input type="checkbox"/>	XS-12 sprint-1 test execution cycle-2	 Sai Ganesh Immidisetti	 Sai Ganesh Immi...	 Medium
<input type="checkbox"/>	XS-11 Test Execution for Test Plan XS-10(sprint-1)	 Sai Ganesh Immidisetti	 Sai Ganesh Immi...	 Medium

Defect Mapping in xray :

it shows **which test caused which bug**, giving full traceability from testing to defects.

Defect mapping in Xray connects **failed tests to bugs**, helping teams track and fix issues efficiently while keeping full visibility inside Jira.

The screenshot shows the Xray interface. At the top, there is a table of defects:

	Work	Assignee	Reporter	Priority	Status
<input type="checkbox"/>	XSP-28 Iteration- and step-level defect 2	Sai Ganesh Immidisetti	Sai Ganesh Immidisetti	Medium	TO DO
<input type="checkbox"/>	XSP-27 Bug found in Exploratory Test XSPTO-8	Sai Ganesh Immidisetti	Sai Ganesh Immidisetti	Medium	TO DO
<input type="checkbox"/>	XSP-26 Iteration- and step-level defect 1	Sai Ganesh Immidisetti	Sai Ganesh Immidisetti	Medium	TO DO

Below the table, a specific defect is selected: "Iteration- and step-level defect 2". The details page shows:

- Key details:** Description
- Action Data Expected Result Actual Result**

Navigate to the store page in Chrome	-	Successful navigation	Landing page is wrong.
Create a profile for user type B	-	Profile is created successfully, and the user is logged in.	User can't login after creation
- Environment:** Add text
- Details:** Assignee: Sai Ganesh Immidisetti; Reporter: Sai Ganesh Immidisetti; Development: Open with VS Code, Create branch, Create commit; Labels: None; Due date: None; Start date: None; Priority: Medium.

Import workitem from csv :Test parameters in Xray help you **run one test multiple times with different data**, improving efficiency and test coverage.

The screenshot shows the "Bulk create: Setup" screen. At the top, there are tabs: **Setup**, **Settings**, **Map fields**, and **Map values**. The **Setup** tab is active.

Setup

To import issues in bulk, you need to provide the data in a CSV file format.

CSV Source File *
 No file chosen
The maximum file upload size is 1,024.00 MB.

Use an existing configuration file
If you have used this importer before, you may have saved the configuration you used.
You can use that configuration again to save time.

Next **Back**

Step 1: Prepare the CSV File

Your CSV should have columns like:

- **Summary** → Test case title
- **Description** → Test case details
- **Issue Type** → Test (or Test Execution if creating executions)
- **Project Key** → Jira project
- **Test Steps** → Steps can be separate columns or formatted properly

Step 2: Go to Jira → Xray Import

1. In Jira, click **Xray** in the top menu
2. Choose **Import Test Cases** or **Test Execution Import**
3. Select **CSV Import**

Step 3: Upload CSV

1. Upload your CSV file
2. Map CSV columns to Jira fields:
 - Summary → Summary
 - Description → Description
 - Test Steps → Test Steps
 - Expected Results → Expected Results
 - Issue Type → Issue Type
 - Project → Project
3. Verify the mapping

Step 4: Configure Test Steps (if needed)

If your CSV contains multiple steps, you may need to **configure step separator** (like ; or |) so Xray reads each step separately.

Step 5: Import & Verify

1. Click **Import**
2. Jira will create Test issues based on CSV
3. Check the imported Test issues to ensure Test Steps and Expected Results are correct

