

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.

### Quick Setup

[Configure Project](#)

[Watch a Quick Xray Demo](#)

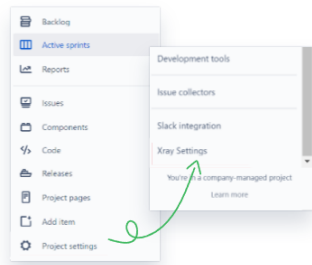
[Contact Support](#)



### Follow these 3 steps to configure Xray

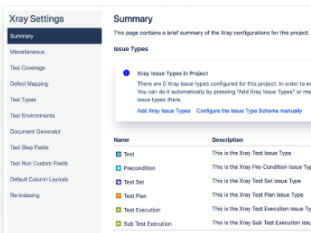
1

From project settings, go to Xray settings.



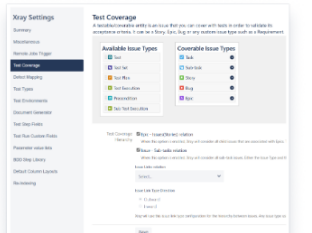
2

Add Xray Issue Types to your project.



3

Set Coverable Issue Types.



\*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

Select parent

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

[Learn about work types](#)

**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 51

**Precondition details**

Precondition details Tests

Manual

Steps

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

Test Repository

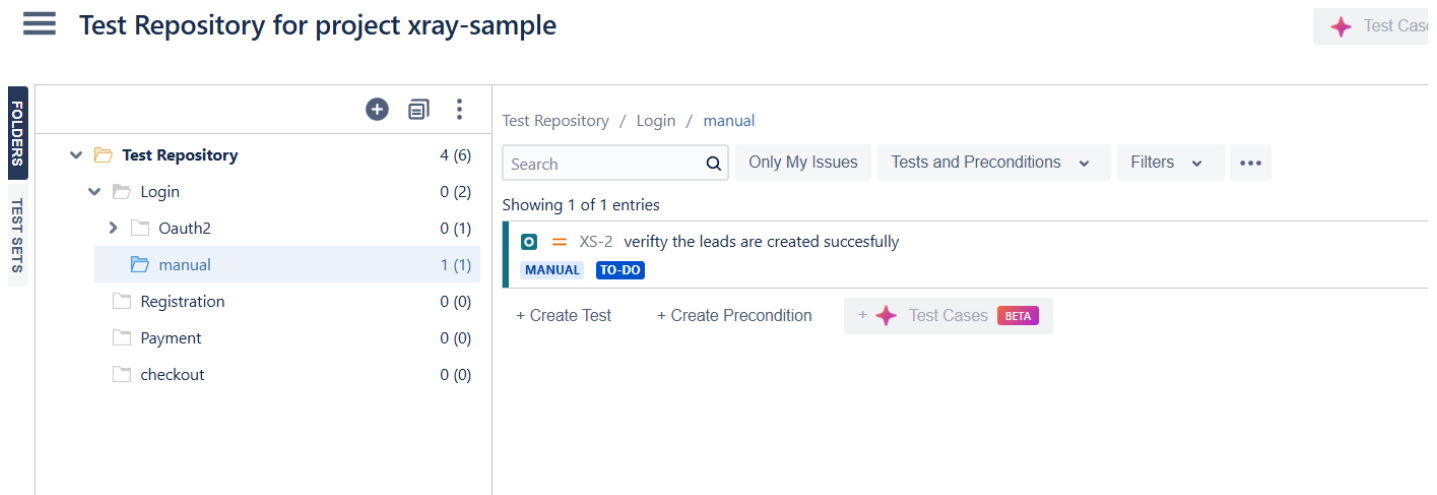
## Test Repository in xray

Go to your Jira project

Click Xray → Test Repository

Create folders

Drag and drop Test issues into folders



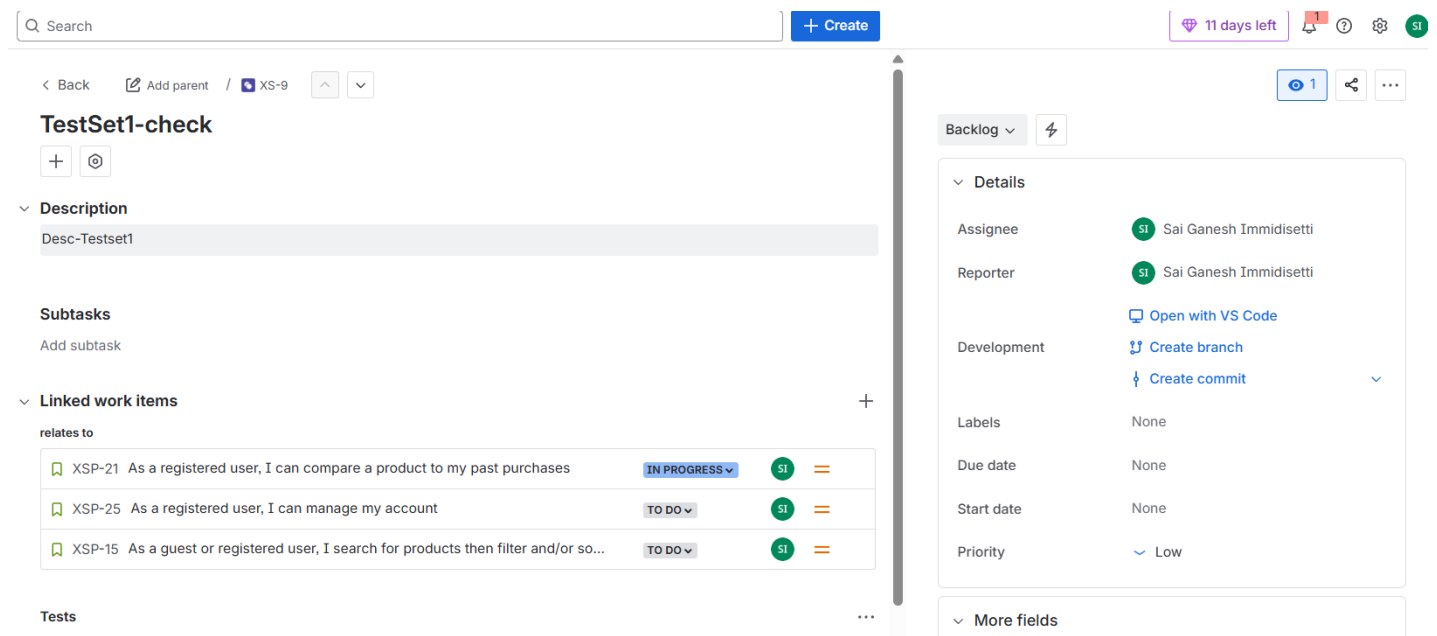
Test Repository only organizes Test issues.

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.

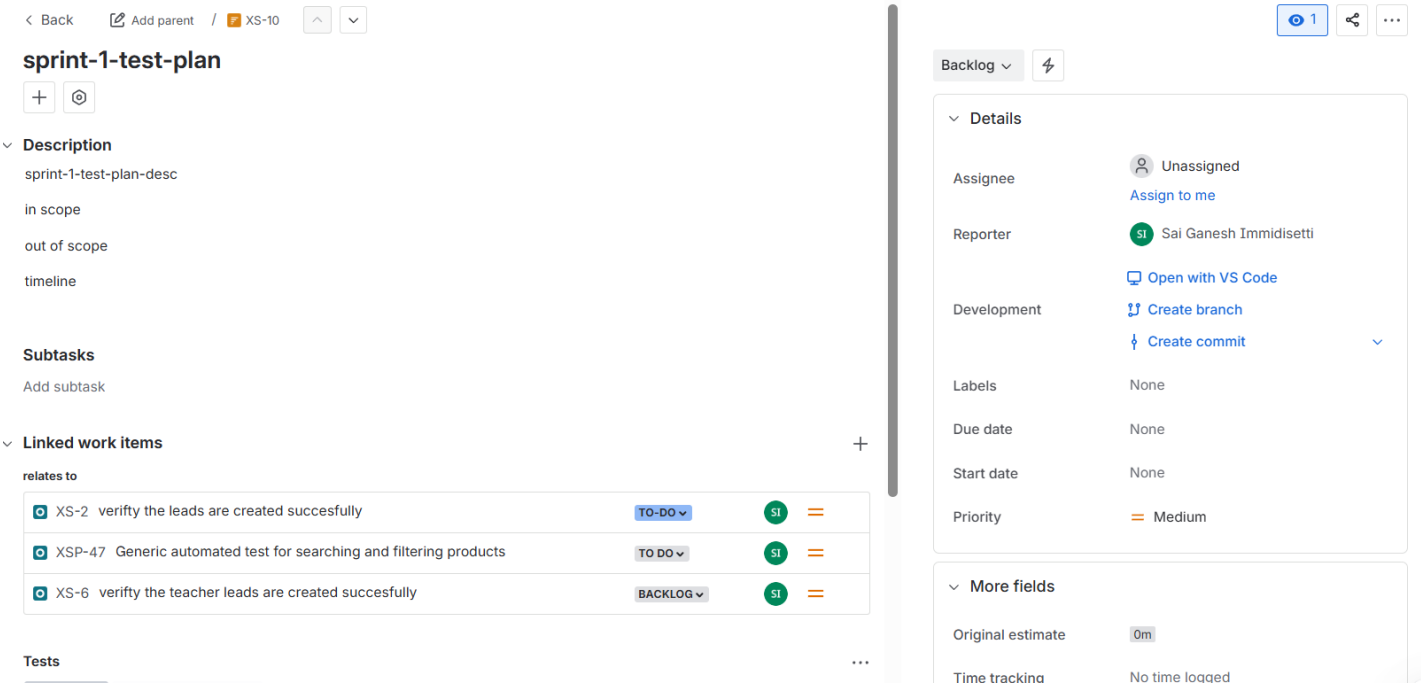


Test Set in Xray = reusable group of test cases.

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.



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}

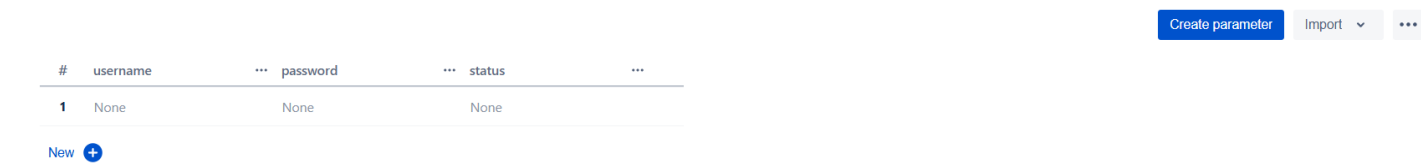
Enter password = \${password}

Execution values:

admin / admin123

user / user123

Dataset for Test XS-2



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.**

Tests

Add Tests

Trigger Build

View on board

Overall Execution Status

1 PASSED

4 TO DO

TOTAL TESTS: 5

Only My Test Runs

Filters

10

Columns

| Rank | Key    | Summary  | Test Type | Dataset | #Defects | TestRun Assignee | Status |
|------|--------|--|-----------|---------|----------|------------------|--------|
| 1    | XS-2   | verify the leads are created succesfully         | Manual    |         | 0        | Unassigned       | PASSED |
| 2    | XS-3   | verify the code works fine                       | Manual    |         | 0        | Unassigned       | TO DO  |
| 3    | XSP-54 | Data-driven manual test for account creation     | Manual    |         | 0        | Unassigned       | TO DO  |
| 4    | XS-6   | verify the teacher leads are created succesfully | Cucumber  |         | 0        | Unassigned       | TO DO  |
| 5    | XS-5   | verify the student leads are created succesfully | Generic   |         | 0        | Unassigned       | TO DO  |

< 1 >

Total 5 issues

Backlog

Details

Assignee: Unassigned  
[Assign to me](#)

Reporter: Sai Ganesh Immidiseti  
[Open with VS Code](#)

Development: [Create branch](#)  
[Create commit](#)

Labels: None

Due date: None

Start date: None

Priority: Medium

Test Plans: [Open Test Plans](#)

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 ☆

| Basic                    | JQL                   | Search work                                  | Space | Assignee              | Type = Test Execution | Status | More filters | Gc |
|--------------------------|-----------------------|--|-------|-----------------------|-----------------------|--------|--------------|----|
| <input type="checkbox"/> | Work                  |  |       |                       |                       |        |              |    |
| <input type="checkbox"/> | <a href="#">XS-13</a> | sprint-1 test execution                      |       | Unassigned            | Sai Ganesh Imm...     | Medium |              |    |
| <input type="checkbox"/> | <a href="#">XS-12</a> | sprint-1 test execution cycle-2              |       | Sai Ganesh Immidiseti | Sai Ganesh Imm...     | Medium |              |    |
| <input type="checkbox"/> | <a href="#">XS-11</a> | Test Execution for Test Plan XS-10(sprint-1) |       | Sai Ganesh Immidiseti | Sai Ganesh Imm...     | 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.

Work

XSP-28

Iteration- and step-level defect 2

XSP-27

Bug found in Exploratory Test XSPTO-8

XSP-26

Iteration- and step-level defect 1

Assignee

Sai Ganesh Immidisetti

Sai Ganesh Immidisetti

Sai Ganesh Immidisetti

Reporter

Sai Ganesh Imm...

Sai Ganesh Imm...

Sai Ganesh Imm...

Priority

Medium

Medium

Medium

Status

TO DO

TO DO

TO DO

< Back

Add parent

XSP-28

Iteration- and step-level defect 2

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

To Do

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.

Bulk create: Setup

Setup

Settings

Map fields

Map values

Setup

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

CSV Source File

Choose 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



