

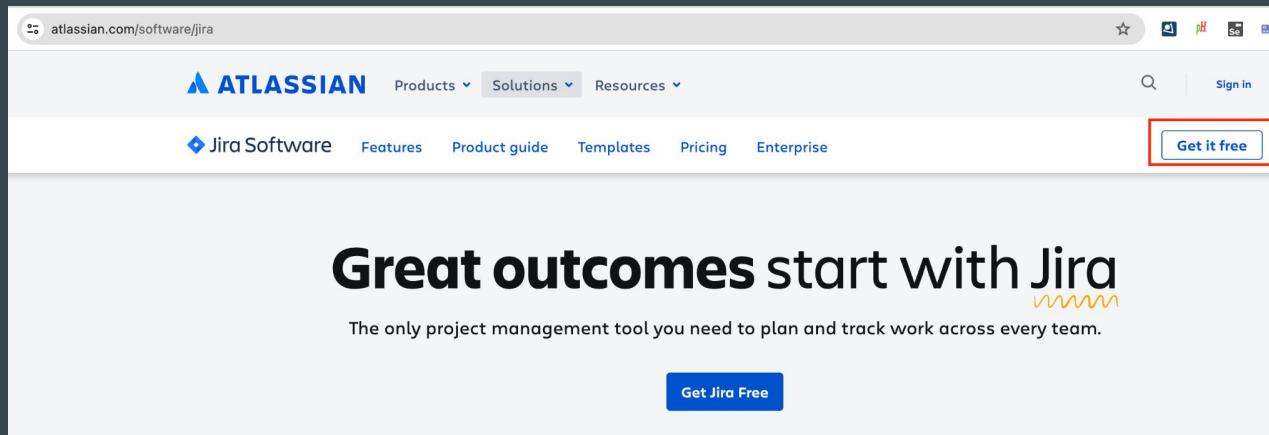
Jira and Xray cloud setup

• • •

Quick setup instructions of a Jira Software + Xray cloud instance

Jira cloud setup

<https://www.atlassian.com/software/jira>



Jira cloud signup

The image consists of three sequential screenshots from the Jira Cloud sign-up process:

- Step 1: Get started with Jira**

Jira logo. Headline: "Get started with Jira". Subtext: "It's free for up to 10 users — no credit card required". Form field: "Work email" containing "testing.uncovered+batman@gmail.com". Agreement text: "Find teammates, plus keep work and life separate by using your own email." TOS and Privacy Policy links.
- Step 2: Verification code sent**

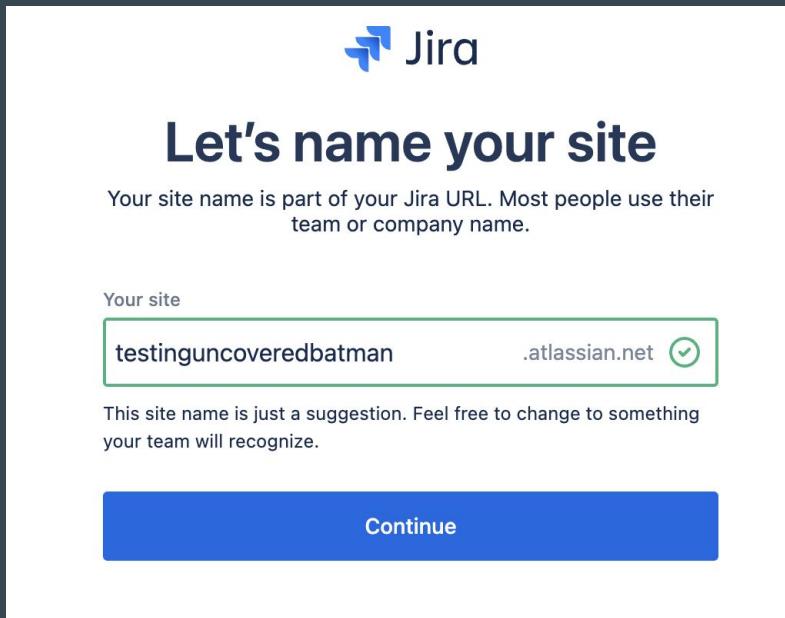
Jira logo. Subtext: "Your verification code is: 513 137". Note: "If you didn't try signing up, you can safely ignore this email". Footer: "This message was sent to you by Atlassian Cloud" and the Atlassian logo.
- Step 3: Enter verification code**

Jira logo. Headline: "We've emailed you a code". Subtext: "To complete your account setup, enter the code we've sent to: testing.uncovered+batman@gmail.com". A row of six input fields contains the code "5 1 3 1 3 |". A "Verify" button is at the bottom, and a "Didn't receive an email? Resend email" link is at the bottom right.

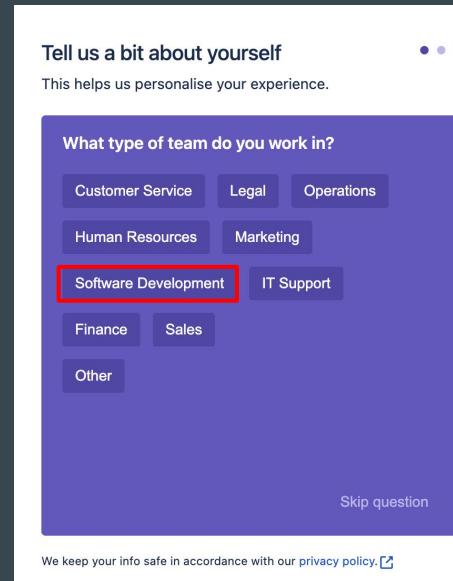
Sergio Freire

Jira cloud signup

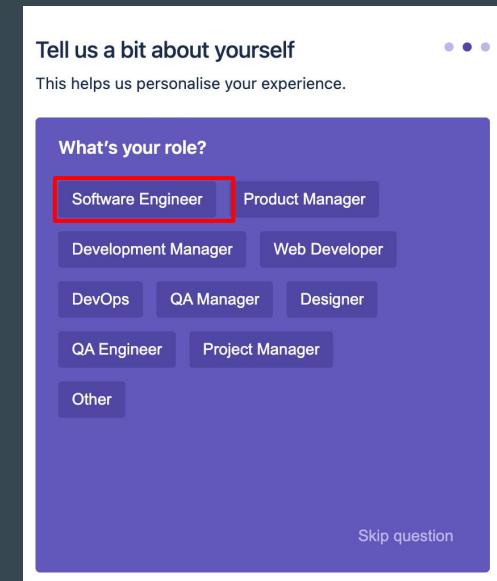
Take note of your Jira Cloud instance URL. You'll be asked some questions that you may skip.



The first step of the Jira Cloud signup process. It features the Jira logo at the top. Below it, the heading "Let's name your site" is displayed in a large, bold, serif font. A sub-instruction below the heading reads: "Your site name is part of your Jira URL. Most people use their team or company name." A text input field contains the placeholder "testinguncoveredbatman" followed by ".atlassian.net" and a green checkmark icon. Below the input field, a note says: "This site name is just a suggestion. Feel free to change to something your team will recognize." At the bottom is a large blue "Continue" button.



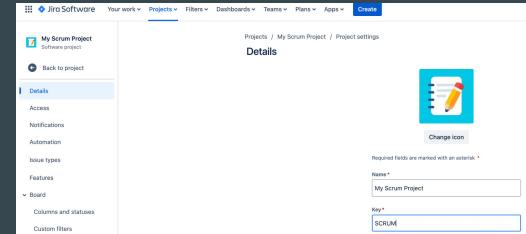
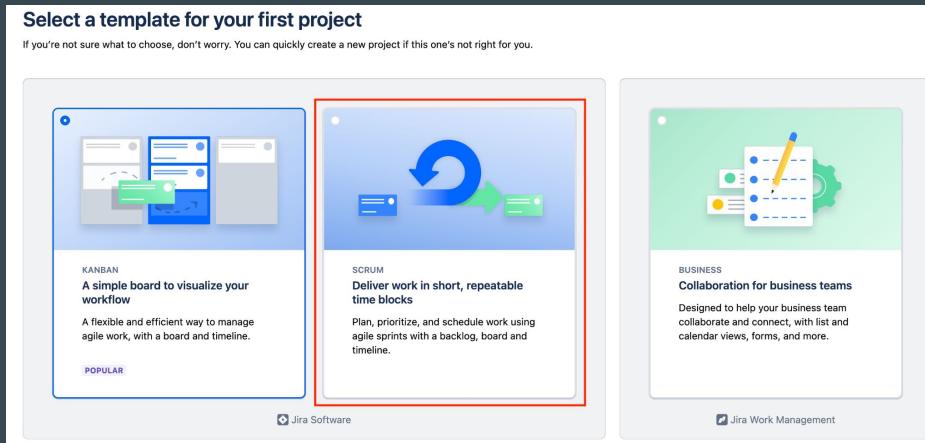
The second step of the Jira Cloud signup process. The heading "Tell us a bit about yourself" is at the top, followed by the sub-instruction "This helps us personalise your experience." There are three dots indicating more steps. The question "What type of team do you work in?" is followed by a grid of buttons: Customer Service, Legal, Operations, Human Resources, Marketing, Software Development (which is highlighted with a red border), IT Support, Finance, Sales, and Other. At the bottom right is a "Skip question" link and a note about privacy: "We keep your info safe in accordance with our [privacy policy](#). 



The third step of the Jira Cloud signup process. The heading "Tell us a bit about yourself" is at the top, followed by the sub-instruction "This helps us personalise your experience." There are three dots indicating more steps. The question "What's your role?" is followed by a grid of buttons: Software Engineer (highlighted with a red border), Product Manager, Development Manager, Web Developer, DevOps, QA Manager, Designer, QA Engineer, Project Manager, and Other. At the bottom right is a "Skip question" link.

Project creation during signup

During the Jira cloud instance signup, you'll be asked to create your first project; this will create a *team-managed* project named “My Scrum Project”, having the key “SCRUM”; therefore, all issues on that project will have keys like SCRUM-<int> (e.g., “SCRUM-1”). You can change the project name and key on the project settings panel.

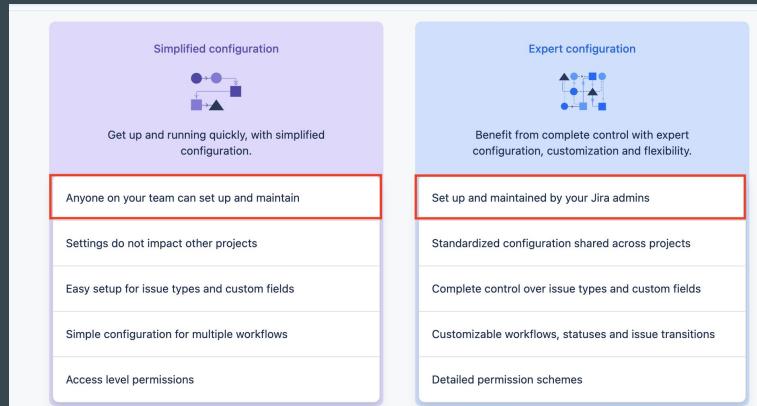


Creation of projects

You can also create your project later on, from the top Projects menu.

Have in mind though that there are 2 main project types in Jira Cloud:

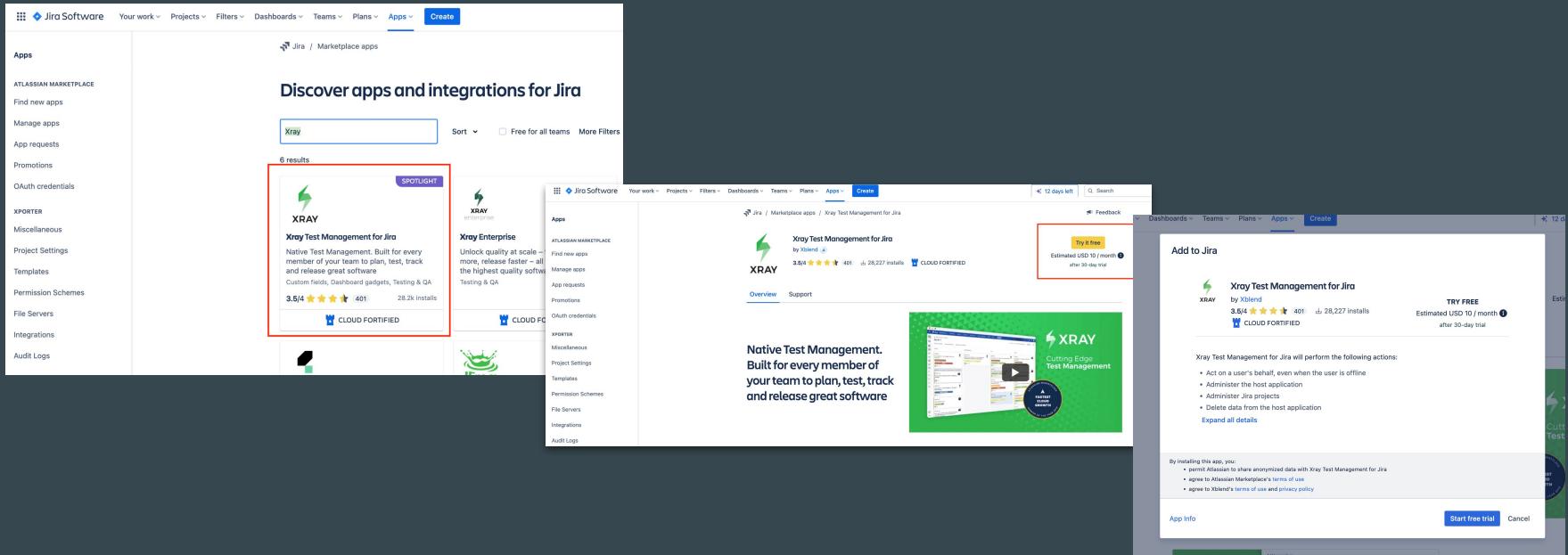
- **Team-managed** (simpler but not so powerful; good enough for us)
- Company-managed (more complex and more powerful)



Xray Installation

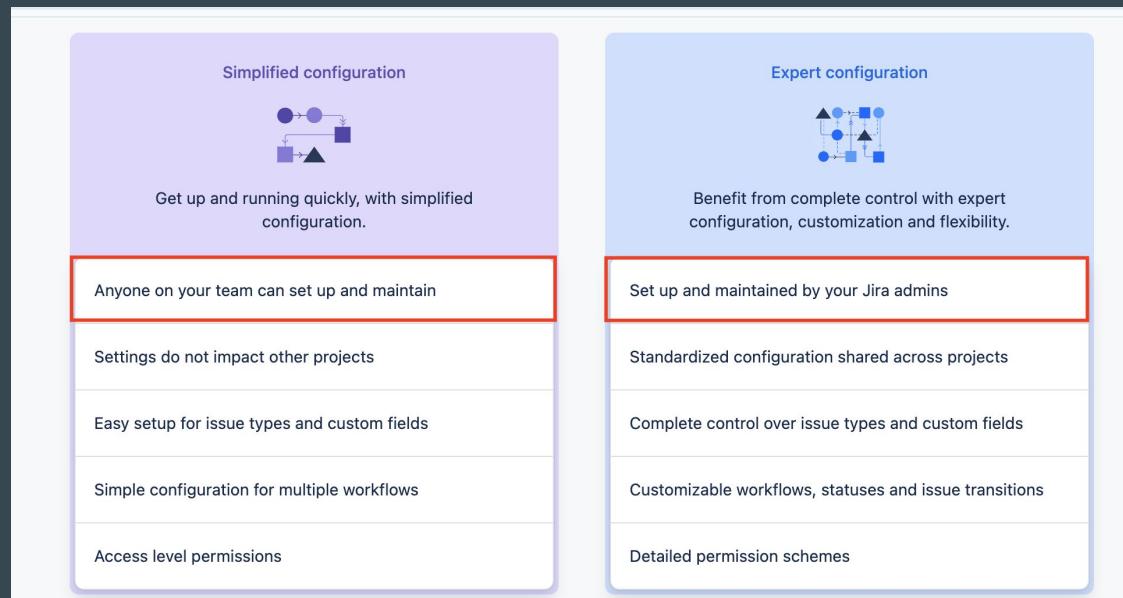
Installation of Xray

Installation is quite straightforward; go to Apps > Explore more apps > Xray and then “Try it free”. You need to use the account that created the Jira instance.



Enabling Xray on Jira projects

- Note: there are 2 main project types in Jira Cloud:
 - Team-managed
 - Company-managed
- We can enable Xray in any of these project types but the configuration steps are slightly different



Organization strategy: “All-in-One”

A single project to manage your “Requirements” (stories, epics), Defects (bugs), and Test related issues and also have all your Test Executions.

The idea is to use one project to manage everything related to it, including testing.



Enabling Xray on team managed projects

Setting a team-managed project in a nutshell

Create or use an existing team-managed project. You can use the project you created during the Jira signup.

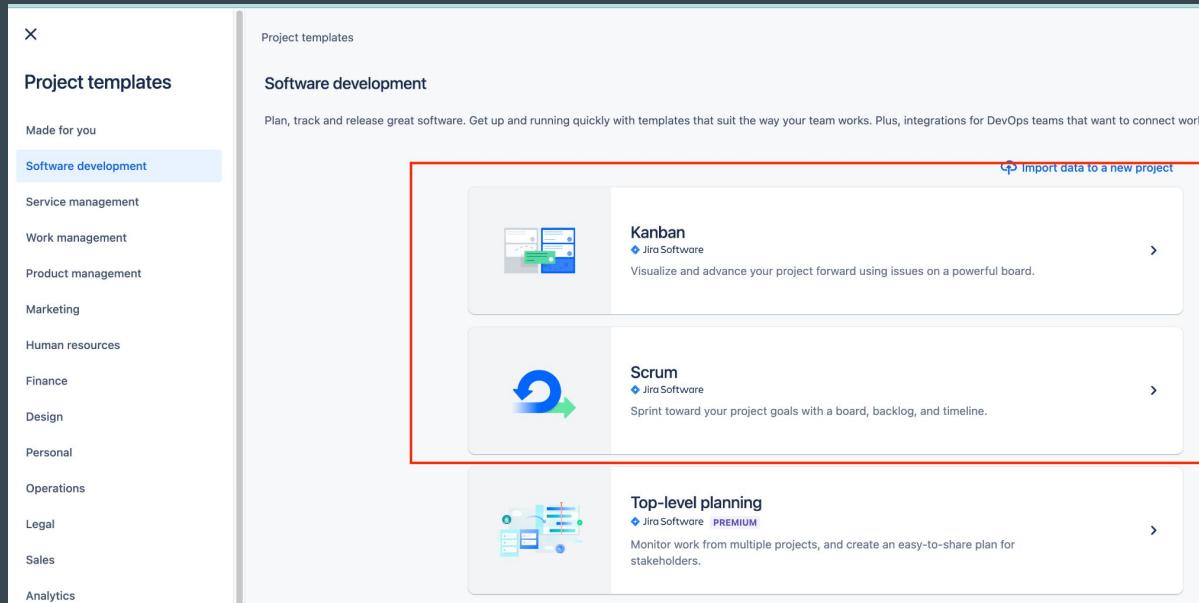
Then:

1. Add (i.e., create) issue types to the project and map them to the Xray concepts
 - a. Precondition, Test, Test Set, Test Execution, Test Plan
2. Define which items you consider to be “requirements” like
3. Define which items you consider to be “defects” like

That's it.

Create project (unless you have created one already)

- Create a Jira project using a Kanban or Scrum template



Create project (unless you have created one already)

- We'll use the Scrum template

Add project details

Explore what's possible when you collaborate with your team. Edit project details anytime in project settings.

Name *

Access *

Open

Key ⓘ *

Template

Scrum

Jira Software

Sprint toward your project goals with a board, backlog, and timeline.

Type

Team-managed

Control your own working processes and practices in a self-contained space.

Cancel Next

Xray project setup

- Go to **Project settings > Apps > Xray Settings**
- There are a few configurations to perform

Jira Software

Projects / SCRUMDEMO

SCRUMDEMO board

TO DO IN PROGRESS DONE

Get started in the backlog

Plan and start a sprint to see issues here.

Go to Backlog

Project settings

Jira Software

Projects / SCRUMDEMO / Project settings / Apps

Xray Settings

Summary

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

| Xray Entity | Issue Type | Description |
|----------------|----------------|-------------|
| Test | Not configured | |
| Precondition | Not configured | |
| Test Set | Not configured | |
| Test Plan | Not configured | |
| Test Execution | Not configured | |

Miscellaneous

This project is currently using the global miscellaneous settings. [Configure](#)

Test Coverage

This project has no issue types mapped as covered. [Configure](#)

Defect Mapping

This project has no issue types mapped as defects. [Configure](#)

Test Types

The Test Types configured for this project are: Manual, Generic, Cucumber. [Configure](#)

Test Environments

This project has 0 Test Environments configured. [Configure](#)

Xray project setup

- Go to **Issue Types Mapping** to configure the issue types
- Then go back to project settings > Issue Types using the “project configuration” link to create the issue types used by Xray

Projects / SCRUMDEMO / Project settings / Apps

Xray Settings

- Summary
- Issue Types Mapping**
- Miscellaneous
- Remote Jobs Trigger
- Test Coverage
- Defect Mapping
- Test Types
- Test Environments
- Document Generator
- Test Step Fields
- Test Run Custom Fields
- Parameter value lists
- BDD Step Library
- Default Column Layouts
- Re-Indexing

Team-Managed Project Issue Types Mapping

The issue type configuration for this team-managed project.

Test

Select an issue type

Precondition

Select an issue type

Test Set

Select an issue type

Test Plan

Select an issue type

Test Execution

Select an issue type

Manual configuration required

In order to configure Xray issue types for this team-managed project, you first need to create the issue types manually on the [project configuration](#).

After the creation of the issue types for the Xray entities you need, you must associate them below.

Projects / SCRUMDEMO / Project settings / Issue types

Jira Software

SCRUMDEMO Software project

Project settings

Issue types

- Epic
- Bug**
- Story
- Task
- Subtask

+ Add issue type

Description fields

Ao Summary

Description

Context fields

Status

Assignee

Labels

Due

Xray project setup: creation of issue types

- Create the following issue types with these names:
 - Precondition, Test, Test Set, Test Plan, and Test Execution
- Optionally: configure the related icons ([download them here](#))

The screenshot shows the Jira Software interface with the 'SCRUMDEMO' project selected. On the left, there's a sidebar with 'Project settings' and a list of existing issue types: Epic, Bug, Story, Task, and Subtask. A red box highlights the '+ Add issue type' button at the bottom of this list. The main content area shows the 'Bug' issue type configuration, including its description ('Bugs track problems or errors.'), required 'Summary' field, and context fields like 'Status', 'Assignee', 'Labels', and 'Report'. There are also 'Edit workflow' and '...' buttons.

The screenshot shows the 'Create issue type' dialog box. It has fields for 'Name*' (with 'Tes' typed in), 'Description' (with placeholder text 'Let people know when to use this issue type'), and an 'Icon' section with a 'Change icon' button. At the bottom are 'Create' and 'Cancel' buttons.

Xray project setup: associate the issue types

This step is for Xray to be aware of the issue types we want to use for its own entities; we could have created “Test Case”, “Test List” issue types (for example) and associate them with the Test and Test Set concept of Xray.

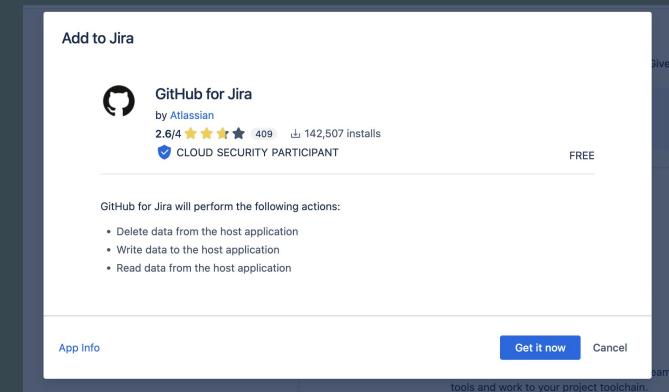
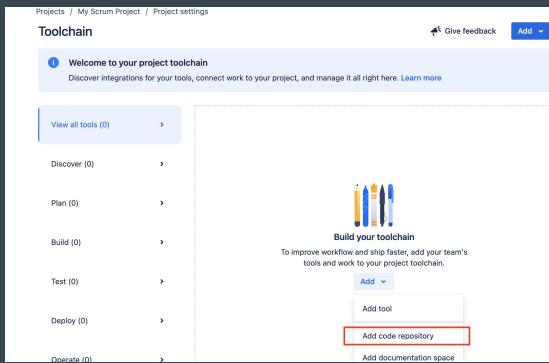
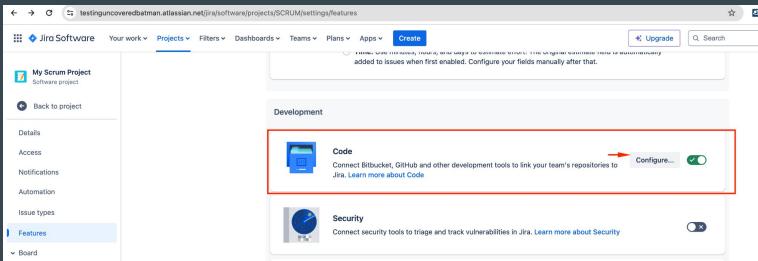
However, in our case, we'll keep it simple, using issue types with the same names of the expected entities:

- Test > Test
- Precondition > Precondition
- Test Set > Test Set
- Test Plan > Test Plan
- Test Execution > Test Execution

The screenshot shows the Jira Software interface for the 'SCRUMDEMO' project. The top navigation bar includes 'Jira Software', 'Your work', 'Projects', 'Filters', 'Dashboards', 'Teams', 'Plans', 'Apps', and a 'Create' button. The 'Projects' dropdown is open, showing 'SCRUMDEMO / Project settings / Apps'. The 'Xray Settings' section is active, with the 'Issue Types Mapping' tab selected. A sidebar on the left contains links for 'Details', 'Access', 'Notifications', 'Automation', 'Issue types', 'Features', 'Board', 'Toolchain', and 'Apps', with 'Xray Settings' highlighted. The main content area is titled 'Team-Managed Project Issue Types Mapping' and describes the configuration for this team-managed project. It lists five mappings: 'Test' (Issue type: Test), 'Precondition' (Issue type: Precondition), 'Test Set' (Issue type: Test Set), 'Test Plan' (Issue type: Test Plan), and 'Test Execution' (Issue type: Test Execution). Each mapping row has a 'Save' button at the bottom.

Nice-2-Have

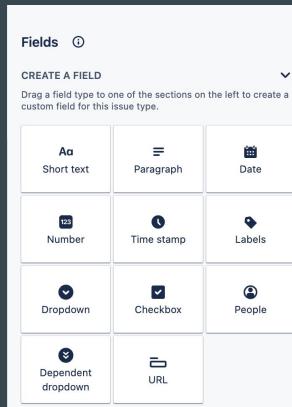
In project settings > **Features**, enable Code and add the code repository tool (e.g., GitHub).



OPTIONAL: Associate some custom fields to Xray issue types

On **Project settings > Issue types**, for the following issue types add the fields

- Test Execution: Revision
 - As short text
- Test Plan: Begin Date and End Date
 - As date



The screenshot shows the 'Issue types' configuration page. Under 'Description fields', there is a 'Summary' field marked as 'REQUIRED'. Under 'Context fields', there is a list of fields: Status, Assignee, Labels, Parent, Story point estimate, Sprint, and Fix versions. The 'Revision' field is highlighted with a red box at the bottom of the list.

Projects / SCRUMDEMO / Project settings / Issue types

Test Execution

Let people know when to use this issue type

Description fields

Aa Summary REQUIRED

= Description

Context fields

→ Status

👤 Assignee

🏷 Labels

📁 Parent

🔢 Story point estimate

📅 Sprint

📌 Fix versions

Aa Revision

HIDE WHEN EMPTY

Xray: define “requirements” / issues coverable by tests

- **Coverable Issue Types:** *Which issue types do you aim to cover with tests? (e.g., Story, Epic)*

The screenshot shows the 'Xray Settings' page in Jira Software. The left sidebar has a 'Test Coverage' section highlighted. The main content area is titled 'Test Coverage' and describes it as an issue type that can be covered with tests. It lists 'Available Issue Types' (Task, Bug, Subtask, Test, Precondition, Test Set, Test Plan, Test Execution) and 'Coverable Issue Types' (Story, Epic). A red box highlights the 'Coverable Issue Types' section. Below it are fields for 'Issue Links relation' (Select...) and 'Default Column Direction' (Outward, Inward), with a note about link type configuration.

Xray: define defects

- **Defect Issue Types:** *Which issue types do you want to be handled as defects? (e.g., Bug)*

The screenshot shows the 'Xray Settings' page in Jira Software, specifically the 'Defect Mapping' section. The left sidebar lists various project settings like Details, Access, Notifications, Automation, Issue types, Features, Board, Toolchain, Apps, App fields, Slack integration, and Xray Settings. The 'Xray Settings' tab is currently selected. The main content area has a header 'Defect Mapping' with a sub-header: 'A defect represents an error, flaw, failure or fault in the SUT (System Under Test) that produces an incorrect or unexpected result. All the issue types mapped as Defect Entities are handled as defects.' Below this, there are two columns: 'Available Issue Types' on the left and 'Defect Issue Types' on the right. The 'Available Issue Types' column contains icons for Story, Task, Epic, Subtask, Test, Precondition, Test Set, Test Plan, and Test Execution. The 'Defect Issue Types' column contains a single item: 'Bug'. At the bottom right of the main area is a 'Save' button.

Xray: final checkup

Go to Project settings > Apps > Xray Settings > Summary.

The screenshot shows the Jira Software interface with the following details:

- Header:** Jira Software, Your work, Projects, Filters, Dashboards, Teams, Plans, Apps, Create, 13 days left, Search.
- Project Information:** SCRUMDEMO, Software project.
- Sidebar:** Details, Access, Notifications, Automation, Issue types, Features, Board, Toolchain, Apps (App fields, Slack integration), Xray Settings (selected).
- Content Area:**
 - Summary:** This page contains a brief summary of the Xray configurations for this project.
 - Issue Types:** Xray Issue Types in Project (5 configured).

| Xray Entity | Issue Type | Description |
|----------------|----------------|-------------|
| Test | Test | |
| Precondition | Precondition | |
| Test Set | Test Set | |
| Test Plan | Test Plan | |
| Test Execution | Test Execution | |
 - Miscellaneous:** This project is currently using the global miscellaneous settings. [Configure](#)
 - Test Coverage:** This project has the following issue types mapped as covered: Story, Epic. [Configure](#)
 - Defect Mapping:** This project has the following issue types mapped as defects: Bug. [Configure](#)
 - Test Types:** The Test Types configured for this project are: Manual, Generic, Cucumber. [Configure](#)
 - Test Environments:** This project has 0 Test Environments configured. [Configure](#)

If all goes well...

- You should see a “Testing Board” shortcut on the side menu
- Whenever creating issues, you should see the issue types that were created earlier

A screenshot of the Jira Software interface for the project 'SCRUMDEMO'. The top navigation bar includes 'Your work', 'Projects', 'Filters', 'Dashboards', 'Teams', 'Plans', 'Apps', and a 'Create' button. The left sidebar has sections for 'Planning', 'Development', and 'Testing'. Under 'Testing', the 'Testing Board' option is highlighted with a red box. The main content area shows the 'Test Repository' for the project, listing one entry: 'SCRUMDEMO-3 manual test for valid login scenario'.

A screenshot of the Jira Software interface showing the 'Create issue' dialog. The 'Create' button in the top right is highlighted with a red box. The dialog has fields for 'Project' (set to 'SCRUMDEMO (SCRUMDEMO)') and 'Issue type'. A dropdown menu under 'Issue type' is open, showing several options: Story, Task, Bug, Epic, Test, Precondition, Test Set, Test Plan, and Test Execution. The 'Test' option is highlighted with a red box. At the bottom of the dialog, there is a note about markdown support and a toolbar for rich text editing.

Try it out, just in case :)

1. Create project with your name in uppercase (e.g., “SCRUMDEMO”); notice the project key (e.g., “SCRUMDEMO”)
2. Create Epic “client area” and a Story “login”; associate Story to the parent Epic
3. Create a Test from the Story screen

You should see coverage information on the Epic and also on the Story issue screen.

This screenshot shows the Jira Story screen for an epic titled "client area". The story key is SCRD-1. The story description is "As a user, I can login the web application". Under the "Test Coverage" section, there is one test named "SCRD-1 manual test for valid login scenario" which is currently in the "TO DO" status. The overall status for the story is "NOTRUN".

This screenshot shows the Jira Story screen for an epic titled "client area". The story key is SCRD-1. The story description is "As a user, I can login the web application". Under the "Test Coverage" section, there is one test named "SCRD-1 manual test for valid login scenario" which is currently in the "TO DO" status. The overall status for the story is "NOTRUN".

This screenshot shows the Jira Test screen for an epic titled "client area". The test key is SCRD-1. The test description is "manual test for valid login scenario". The test status is "TO DO". The test details section indicates that there are no steps defined. A note at the bottom states: "A test is a sequence of steps coupled with conditions, test inputs and expected results. Please create or import test steps to define the test".

Xray Entities & Concepts

...

Xray entities & concepts (a few)

Xray provides several concepts; we'll focus just on a few ones.

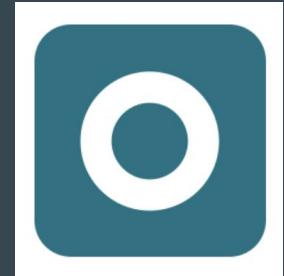
- Test
- Test Run
- Test Execution
- Test Plan
- “Requirements”
- “Defects”

All of the previous entities will be issues in Jira/Xray, except the Test Run which is

Test

An abstraction of a test idea/scenario, automated test, and thus reusable.

It is essentially a way to verify/validate the associated requirement(s).



- Can be a scripted (e.g. test case, automated test) or exploratory test
- Can be specified using Gherkin (Scenario)
- Can be executed manually in Jira/Xray or through automation
- May be linked to/cover 1 (or more) requirements, using the “tests” issue link type
- Has one type: “Manual”, “Cucumber”, “Generic”

Test: Manual

A traditional test case composed by a list of steps, thus scripted.

- Each step is mainly defined by:
 - Action/step
 - Expected result

The screenshot shows the Xray software interface for managing test cases. The main area displays a 'Test details' card for a 'Manual' test type. The test consists of three steps:

| Step | Action | Data | Expected Result |
|------|---------------------|------|-----------------|
| 1 | Action press 1 | None | 1 |
| 2 | Action press key | + | + |
| 3 | Action press 2 | None | 2 |

On the right side, there is a detailed sidebar with various fields for the test case, including:

- To Do: Unassigned (Assign to me)
- Reporter: Sérgio Freire
- Development: Create branch, Create commit
- Releases: None
- Labels: None
- Revision: None
- ExternalRef: None
- Priority: Medium
- Xporter: Open Xporter
- Test Status: Open Test Status
- More fields: Original estimate, Time tracking, Epic Link, Components, Sprint...
- Automation: Rule executions

At the bottom, it shows the creation and update times: Created March 9, 2022 at 5:53 PM and Updated March 9, 2022 at 5:53 PM.

If we don't have test automation yet, or the test scenario is hard/costly to automation, we can create a “manual” test and link it to the related Story, for example. We can then manually record its result in Xray.

Test: Cucumber

A Cucumber Scenario/Scenario Outline that provides one or more examples of an acceptance criteria.

- Write tests in a business-readable domain-specific language (Gherkin)
- Specify Cucumber Scenarios and Scenario Outlines with Gherkin syntax highlighting
- Ability to export to .feature files and execute during Continuous Integration

Test Type
Cucumber ▾

Scenario

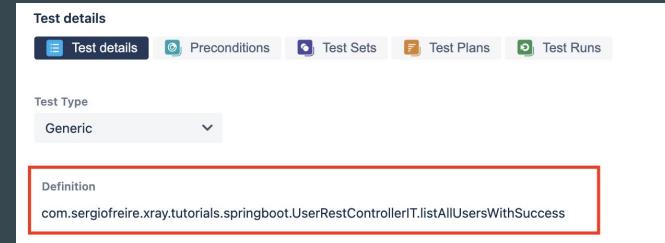
```
1 Given I have entered "<input_1>" into the calculator
2 And I have entered "<input_2>" into the calculator
3 When I press "<button>"
4 Then the result should be "<output>" on the screen
5
6 Examples:
7 | input_1 | input_2 | button | output |
8 | 20     | 30      | add    | 50   |
9 | 2      | 5       | add    | 7    |
10 | 0     | 40      | add    | 40   |
11 | 4     | 50      | add    | 54   |
12 | 5     | 50      | add    | 55   |
```

We can use Xray as the master for the Cucumber/Gherkin scenarios or we can use Git instead. The flows for fully integrating the results from automation are slightly different as shown in [this tutorial](#).

Test: Generic

A way to abstract and have visibility of traditional automated tests or exploratory tests.

- Allows you to track the results of automated tests that are non-Cucumber (e.g. JUnit)
 - Automate tests in any framework and report results back to Xray
 - These tests are usually auto-provisioned whenever importing the results the first time
 - Definition field contains a unique identifier of automated test like
<package_name>.<class_name>.<method_name>



Note: Generic tests could also be used to have visibility of Cucumber results if upload JUnit XML reports. In that case we lose visibility of the individual Gherkin sentences; we would just have visibility of the whole test result.

Test Execution

A “task” for executing a group of tests on a given version of the system, on a given environment. This task will also contain the results when they’re reported. Sometimes the Test Execution is created alongside with its results (i.e., the Test Runs).

- Contains a list of Tests and their results (i.e. Test Runs)
- May be planned (especially for manual tests) or ad hoc
- May be created manually (i.e., “I want to run these tests...”);
 - Later on someone will “execute” it manually, by going to each test and record the actual results and whether the test passed or failed
- May be created during Continuous Integration
 - in this case it will already contain the results (i.e., a Test Run for each Test that was executed)



Test Run

An instance of a Test in the context of a Test Execution. A run of a Test in some scope.

- Contains the results obtained for a Test, including evidence and linked/reported defects
- For compliance, also contains a copy of the Test specification at the moment of execution
- It's an internal entity; not a Jira issue

Test Plan

A way to define the scope for testing and track its progress; what testing we'll be performing and its latest status.



- Tracks a group of tests and their results independently of the number of executions
- Can be used in a planned way, with its scope for testing (i.e. the Tests) defined beforehand
- Can be used in an Agile way, acting as a testing guidance result aggregator, by allowing you to create/add Tests along with their results at any time
- Test Plans may be assigned to versions, sprints and users, as they're a Jira issue
- A version or a sprint may have multiple Test Plans

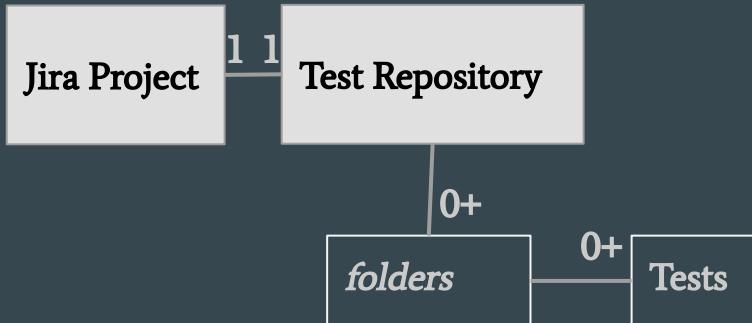
Relation between entities

- “Requirements” are coverable by one or more Tests
- A Test Plan has a list of Tests it tracks and usually several Test Executions related to those Tests
- A Test Execution contains Test Runs, one for each Test that is on the Test Execution

Additional entities... (non issue type based)

Test Repository

- The place where we can see all Tests in a project, organized in folders
- No information about results
- Mostly useful for manual testing



The screenshot shows the 'Test Repository' page for the 'Calculator' project. The left sidebar has tabs for 'FOLDERS' and 'TEST SETS'. The main area displays a tree view of test sets and their counts:

| Folder/Test Set | Count |
|-----------------------------|-------------|
| Test Repository | 1909 (1933) |
| Performance testing | 0 (0) |
| Postman F1 | 14 (14) |
| Postman F2 | 1 (1) |
| core | 1 (1) |
| UI | 0 (8) |
| login | 1 (1) |
| basic arithmetic operations | 7 (7) |

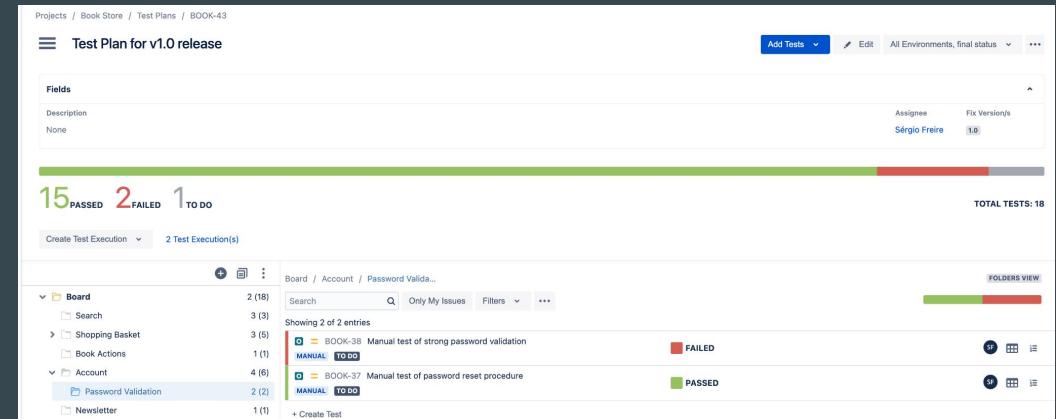
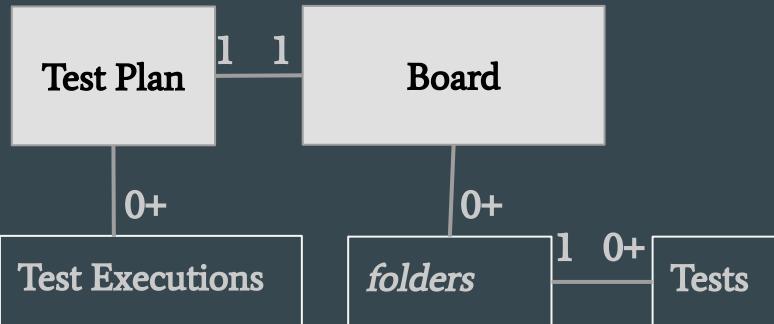
Below the tree view is a table of test cases:

| Test Case | Status |
|--------------------------------------|----------------|
| CALC-2706 dummy gherkin precondition | CUCUMBER TO DO |
| CALC-2666 xx | CUCUMBER TO DO |
| CALC-1450 organizations exist | CUCUMBER TO DO |
| CALC-864 Background for: CALC-861 | CUCUMBER TO DO |
| CALC-393 Calculator exists | CUCUMBER TO DO |

Additional entities... (non issue type based)

Test Plan' Board

- An implicit structure in each Test Plan to organize the Tests in folders, considering priorities and what makes sense from a execution perspective
- Can be useful to group the results logically, using folders



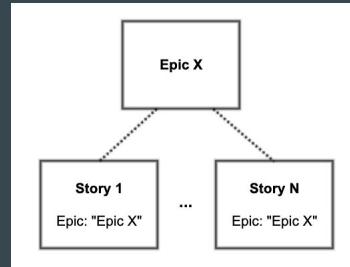
Defects (e.g., “Bug”)

A defect is something that negatively impacts quality (the value perceived to some stakeholder).

- Xray uses the “defect” word in a broader sense, i.e. some deficiency/imperfection related to the product. Usually, these are also known as bugs
- In practice, a “defect” is something we report manually during testing, or after testing whenever analyzing the test results, as a Jira issue (e.g., “Bug”)
- You can define one or more issue types to be treated as “defects”; they can include Bug or any other custom issue types you may want for that purpose

“Requirements” (e.g., Story, Epic)

- Xray uses “requirement” word in broad sense: something that the SUT must meet
- In practice, a “requirement” is any issue in Jira that can be covered (i.e., verified or validated) with Tests
- You can define the issue types to be treated as “requirements”; they can include Story, Epic, or any other custom issue type
- Xray is aware of the hierarchical relation between Epics and Stories; if you create a Test for a Story, you can see it at the related Epic level (i.e., covering the Epic)



Coverage in Xray

Xray provides an heuristic, called (test) coverage; sometimes people may refer to it as requirement coverage, depending on the perspective.

This is different from code coverage, which is focused on covering code.

Coverage in Xray is at higher level; it's a way to understand if a deliverable/requirement (Story, Epic) has some tests associated and if so, if all those tests have been run and were successful.

Coverage: UNCOVERED

In Xray a “requirement” (Story, Epic) is UNCOVERED if it has no Tests linked/covering it.

The screenshot shows a requirement card in the Xray application. At the top, there are navigation links: Projects / Calculator / Add parent / CALC-2855. Below these are several buttons: Attach, Create subtask, Link issue, Test Coverage (which is highlighted with a green icon), and a three-dot menu. The main title of the requirement is "As a user, I can login the web application". Underneath the title are buttons for Attach, Create subtask, Link issue, and a dropdown menu. The "Test Coverage" button is also present here. A "Description" section follows, with a placeholder "Add a description...". Below this is a "Test Coverage" section, which contains a small icon of a clipboard with a magnifying glass over it. A red rectangular box highlights a blue button labeled "UNCOVERED". Below this button is the text "No Tests are associated with this issue." At the bottom of the card is a blue "Add Tests" button with a dropdown arrow.

Coverage: NOTRUN

In Xray, a requirement is “NOTRUN” if:

- *any* of the linked tests need yet to be run

Remember: it's an heuristic! Tests don't cover everything and not every test (problems found during their execution) has the same importance.

The screenshot shows a requirement in Xray with the key 'CALC-2855' and the title 'As a user, I can login the web application'. Below the title are buttons for 'Attach', 'Create subtask', 'Link issue', 'Test Coverage', and more. The 'Test Coverage' section includes a 'Linked issues' table and a 'Test Coverage' table. The 'Test Coverage' table has a yellow bar at the top labeled 'NOTRUN'. The 'Test Status' column shows 'TO DO' for all tests. The 'Summary' column indicates that the test for successful login is 'test for successful login'.

| Status | Key | Summary | Test Status |
|--------|-----------|---------------------------|-------------|
| TO DO | CALC-2856 | test for successful login | TO DO |

Coverage: OK

In Xray, and also in some tools, a requirement is “OK” if ...

- All linked tests passed, according with latest results

Remember: it's an heuristic! Tests passing don't mean we can ensure the requirement doesn't have problems.

The screenshot shows a requirement detail page for 'CALC-2855'. At the top, there are buttons for 'Attach', 'Create subtask', 'Link issue', 'Test Coverage' (which is highlighted in green), and more. Below this is a 'Description' section with a placeholder 'Add a description...'. Under 'Linked issues', it says 'is tested by' and lists 'CALC-2856 test for successful login' with a status of 'TO DO'. In the 'Test Coverage' section, there are buttons for 'Add Tests' and 'Execute'. Below this is an 'Analysis & Scope' table with a single row for 'CALC-2856' with a summary 'test for successful login'. The 'Status' column shows 'TO DO' and the 'Test Status' column shows 'PASSED'. A large green 'OK' button is at the bottom right.

Coverage: NOK

In Xray, a requirement is “NOK” if ...

- *any* of the linked failed, according with latest results

Remember: it's an heuristic! Tests don't cover everything and not every test (problems found during their execution) has the same relevance.

The screenshot shows the Xray application interface for a requirement titled "As a user, I can login the web application".

Linked issues: The requirement is tested by two issues: CALC-2856 (test for successful login) and CALC-2858 (test for invalid login). Both issues are currently marked as "TO DO".

Test Coverage: The analysis scope is set to "Latest Final Status". There are two tests listed:

| Status | Key | Summary |
|--------|-----------|---------------------------|
| TO DO | CALC-2856 | test for successful login |
| TO DO | CALC-2858 | test for invalid login |

A red box highlights the "NOK" status indicator at the bottom right of the coverage table. A legend indicates that green means "PASSED" and red means "FAILED".

OTHER, OPTIONAL INFO

• • •

Just in case you need it...

Enabling Xray on company managed projects

Setting a company-managed project in a nutshell

Create or use an existing company-managed project and then:

1. Add issue types to the project and map them to the Xray concepts
2. Define which items you consider to be “requirements” like
3. Define which items you consider to be “defects” like

That's it.

There are additional settings that we can fine tune ahead.

Create project

- Create a Jira project using a Kanban or Scrum template

The screenshot shows the Jira Project Templates interface. On the left, a sidebar lists categories: Made for you (Software development, Service management, Work management, Product management, Marketing, Human resources, Finance, Design, Personal, Operations, Legal, Sales, Analytics), and Other (Analytics). The 'Software development' category is selected and highlighted with a blue background. The main area is titled 'Project templates' and 'Software development'. It contains three cards:

- Kanban** (Jira Software) - Visualize and advance your project forward using issues on a powerful board.
- Scrum** (Jira Software) - Sprint toward your project goals with a board, backlog, and timeline.
- Top-level planning** (Jira Software PREMIUM) - Monitor work from multiple projects, and create an easy-to-share plan for stakeholders.

A red box highlights the Kanban and Scrum cards. At the top right of the main area, there is a link: [Import data to a new project](#).

Select type of project

| Team-managed | Company-managed |
|---|--|
| <p>Set up and maintained by your team.</p> <p>For teams who want to control their own working processes and practices in a self-contained space. Mix and match agile features to support your team as you grow in size and complexity.</p> <p>Simplified configuration</p>  <p>Get up and running quickly, with simplified configuration.</p> <ul style="list-style-type: none">Anyone on your team can set up and maintainSettings do not impact other projectsEasy setup for issue types and custom fieldsSimple configuration for multiple workflowsAccess level permissions <p>Essential features</p>  <p>A modern Jira experience for teams who don't need advanced features.</p> <ul style="list-style-type: none">Only show your project's issues on your boardEssential agile reporting | <p>Set up and maintained by your Jira admins.</p> <p>For teams who want to work with other teams across many projects in a standard way. Encourage and promote organizational best practices and processes through a shared configuration.</p> <p>Expert configuration</p>  <p>Benefit from complete control with expert configuration, customization and flexibility.</p> <ul style="list-style-type: none">Set up and maintained by your Jira adminsStandardized configuration shared across projectsComplete control over issue types and custom fieldsCustomizable workflows, statuses and issue transitionsDetailed permission schemes <p>Advanced features</p>  <p>All the power and features that Jira Software is known for.</p> <ul style="list-style-type: none">Pull in issues from other projects on your boardComprehensive agile reporting |
| Select a team-managed project | Select a company-managed project |

Add project details

Add project details

Explore what's possible when you collaborate with your team. Edit project details anytime in project settings.

Required fields are marked with an asterisk *

Name *

SAMPLEPROJ

Key ⓘ *

SAM

Share settings with an existing project

Template

Change template



Scrum

◆ Jira Software

Sprint toward your project goals with a board, backlog, and timeline.

Type

Change type



Company-managed

Work with other teams across many projects in a standard way.

Cancel

Next

Xray project setup

- Go to **Project settings > Apps > Xray Settings**
- There are several configurations to perform

The screenshot shows the Jira Software Project settings interface for a project named "SAMPLEPROJ". The left sidebar lists various project settings categories. The "Xray Settings" section is highlighted with a red box. Within "Xray Settings", the "Summary" tab is selected, also highlighted with a red box. The main content area displays a summary of Xray configurations, including sections for "Issue Types", "Miscellaneous", "Test Coverage", "Defect Mapping", and "Test Types". A modal window titled "Xray Issue Types in Project" is open, showing a table of issue types present in the project. The "Issue Types" table has columns for Name, Description, and Present in Project. The "Miscellaneous" section notes the use of global settings. The "Test Coverage" and "Defect Mapping" sections indicate no mappings are defined. The "Test Types" section shows the configured test types: Manual, Generic, and Cucumber.

| 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. | ● |

Xray: define “requirements” / the coverage

- **Coverable Issue Types:** *Which issue types do you aim to cover with tests? (e.g., Story, Epic)*

The screenshot shows the 'Xray Settings' interface with the 'Test Coverage' tab selected. On the left, a sidebar lists various settings like 'Miscellaneous', 'Remote Jobs Trigger', 'Defect Mapping', etc. The main area is titled 'Test Coverage' with a sub-section 'Available Issue Types'. A list of items includes Task, Sub-task, Bug, Test, Test Set, Test Plan, Test Execution, Precondition, and Sub Test Execution. To the right, a section titled 'Coverable Issue Types' lists 'Story' and 'Epic'. An arrow points from the 'Story' checkbox in the 'Available Issue Types' list to the 'Coverable Issue Types' list. Below these sections, there are configuration options for 'Test Coverage Hierarchy' (with 'Epic - Issues(Stories) relation' checked), 'Issue - Sub-tasks relation' (also checked), and 'Issue Links relation' (with a dropdown set to 'Select...'). At the bottom, a note about issue link type direction (Outward or Inward) and a 'Save' button are visible.

Xray: define defects

- **Defect Issue Types:** *Which issue types do you want to be handled as defects?*

(e.g., Bug)

The screenshot shows the 'Xray Settings' interface with the 'Defect Mapping' tab selected. On the left, a sidebar lists various settings like 'Summary', 'Miscellaneous', 'Remote Jobs Trigger', 'Test Coverage', 'Defect Mapping' (selected), 'Test Types', 'Test Environments', etc. The main area is titled 'Defect Mapping' with a sub-instruction: 'A defect represents an error, flaw, failure or fault in the SUT (System Under Test) that produces an incorrect or unexpected result. All the issue types mapped as Defect Entities are handled as defects.' It displays two columns: 'Available Issue Types' (Task, Sub-task, Story, Epic, Test, Test Set, Test Plan, Test Execution, Precondition, Sub Test Execution) and 'Defect Issue Types' (Bug). A red arrow points from the 'Available Issue Types' column to the 'Bug' entry in the 'Defect Issue Types' column.

Xray: final checkup

Go to Project settings > Apps > Xray Settings > Summary.

The screenshot shows the 'Xray Settings' page under the 'SAMPLEPROJ' project. The left sidebar lists various project settings like Summary, People, Permissions, Notifications, Automation, Features, Toolchain, Workflows, Issues, Types, Layout, Screens, Fields, Collectors, Security, Components, Apps, Microsoft Teams Integration, Slack integration, and Development tools. The 'Xray Settings' section is highlighted. The main content area is titled 'Summary' and contains a brief description of Xray configurations. It features a 'Issue Types' section with a message about 6 configured issue types, a table listing 8 issue types with descriptions and status, and sections for Miscellaneous, Test Coverage, Defect Mapping, Test Types, and Test Environments.

Issue Types

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. | ✓ |

Miscellaneous

This project is currently using the global miscellaneous settings. [Configure](#)

Test Coverage

This project has the following issue types mapped as covered: Story, Epic. [Configure](#)

Defect Mapping

This project has the following issue types mapped as defects: Bug. [Configure](#)

Test Types

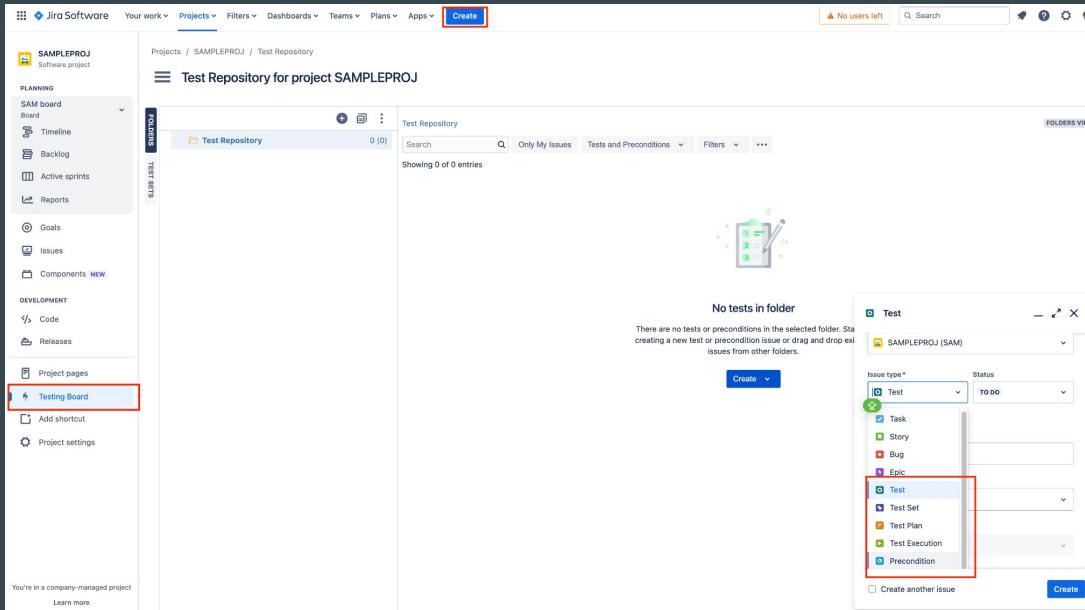
The Test Types configured for this project are: Manual, Generic, Cucumber. [Configure](#)

Test Environments

This project has 0 Test Environments configured. [Configure](#)

If all goes well...

- You should see a “Testing Board” shortcut on the side menu
- Whenever creating issues, you should see the issue types that were created earlier



DEMO / Try it out, just in case :)

1. Create Epic and a Story; associate Story to the parent Epic
2. Create a Test from the Story screen

You should see coverage information on the Epic and also on the Story issue screen.

The image displays three screenshots of a software application interface, likely Jira, illustrating the creation of a project structure and associated tests.

Screenshot 1: Project Overview (SAM-1)

This screenshot shows the main project page for "SAM-1". It includes sections for "client area", "Description", "Child issues" (listing "SAM-2"), and "Test Coverage". The "Test Coverage" section shows a table with one row:

| Status | Key | Summary | Test Status |
|--------|-------|-------------------------------|-------------|
| TO DO | SAM-3 | test for valid login scenario | TO DO |

Screenshot 2: Story Details (SAM-1 / SAM-2)

This screenshot shows the story details for "As a user, I can login the web application" under "SAM-1 / SAM-2". It includes sections for "Description", "Linked issues" (listing "SAM-3"), and "Test Coverage". The "Test Coverage" section shows a table with one row:

| Status | Key | Summary | Test Status |
|--------|-------|-------------------------------|-------------|
| TO DO | SAM-3 | test for valid login scenario | TO DO |

Screenshot 3: Test Creation (SAM-3)

This screenshot shows the test creation screen for "test for valid login scenario" under "SAM-3". It includes sections for "Description", "Linked issues" (listing "SAM-2"), and "Test details". The "Test details" section shows a table with one row:

| Test Type | Test Status |
|-----------|-------------|
| Manual | NOTRUN |

The "Test steps" section indicates "There are no steps defined".