

Tips for a successful migration path from TestRail Lite to Xray

- Overview
- Installation and configuration
- Testing activities
 - Specification
 - TestRail Lite
 - Xray
 - Organization
 - TestRail Lite
 - Xray
 - Planning
 - TestRail Lite
 - Xray
 - Execution
 - TestRail Lite
 - Xray
 - Reporting
 - TestRail Lite
 - Xray
- Moving forward

Overview

This article provides overall guidance on migrating from TestRail Lite to Xray, so you can take advantage of Xray unique capabilities.

While TestRail Lite is focused on simplicity and targeted for manual testing, Xray is a more complete and flexible solution that can also embrace test automation and exploratory testing.



TestRail vs TestRail Lite

TestRail is a Test Management solution available as a standalone application; TestRail Lite is a lighter and tailored version for Jira.

If you've used TestRail Lite before, adopting Xray is easy.

In TestRail Lite, a user creates some Test Cases in a requirement/user story and then executes them by creating a Test Run. In Xray we can do the same but we have additional possibilities, related to specification, organization, planning, and reporting.

But before getting into details, we need to clarify some concepts.

The following table sums up a set of common Test Management functionalities and how those can be addressed using TestRail Lite and on Xray.

There are some noticeable differences, namely on the organization level where Xray is more flexible by providing an hierarchical organization for each project; Xray also provides a way to organize Tests in flat lists using Test Sets.

In terms of planning, Xray has a dedicated entity for this purpose: the Test Plan.

Also worth highlighting is the Test Run concept which exists in both tools but has different meanings.

Functionality	TestRail Lite	Xray	Notes
Specification	Test Cases, Precondition (as a field in the Test Case)	Tests, Precondition (as Jira issue that can be linked to Tests)	Test Cases will have a 1:1 match with Xray' Tests. The same occur with Preconditions; however, in Xray, Preconditions are a dedicated entity and reusable between Tests.
Organization	Test Suite + Test Section	Test Repository, Test Set	Test Suite and Sections will match the Test Repository concept. In Xray, each project has its own Test Repository for organizing tests in folders.
Planning	-	Test Plan	TestRail Lite doesn't provide any planning capabilities; eventually, Test Suite and Sections can be used even though they're not tailored for that. In Xray we have an entity to serve this purpose: Test Plan, although it is not mandatory we encourage its usage.
Execution	Test Run (and corresponding Test Results)	Test Execution (and corresponding Test Runs)	TestRail Lite's Test Run will be translated into an Xray Test Execution. Each (Test) Result will be translated into a Xray' Test Run.

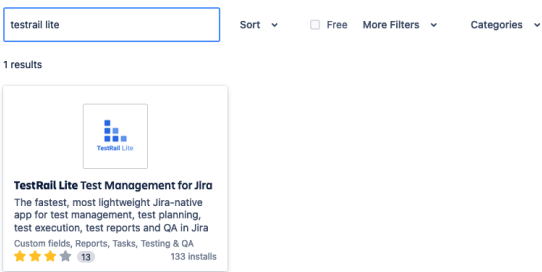
Reporting	Reports	Coverage, Traceability (others)	TestRail Lite only provides a simple Test Runs view. On the other hand, in Xray, you will see (at least) a set of Built-in Reports that will help you during your process.
-----------	-------------------------	---	--

In the next sections we will go over typical usage scenarios, including the installation and configuration process, so we can understand how these can be achieved in both tools.

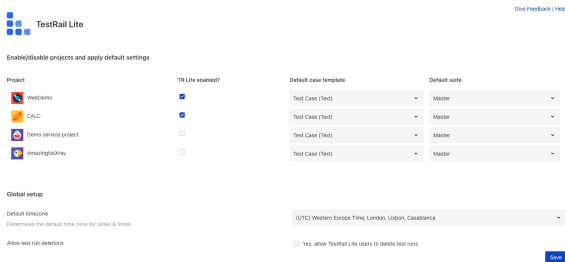
Installation and configuration

In order to install TestRail Lite you just have to search the app and install it through the Apps entry of Jira.

Discover apps and integrations for Jira



TestRail Lite will be available for all your projects and you can choose to enable/disable it from the configuration page, on a per project basis, and apply the default settings regarding the suite and the case template to be used by default.



Other than the option to active TestRail Lite in each project you can also define which is the default case template to be used or the default suite (it comes by default with a Master suite).

On the Global Setup area you can define the default timezone to be used and if you allow TestRail Lite user to delete test runs.

All these new objects (cases, suites, test runs) are internal to TestRail Lite, meaning that you cannot search, use in pre defined workflows or define a custom field.

It also means that they will not appear in your board or reports, they only will surface in the context of the Jira issues.

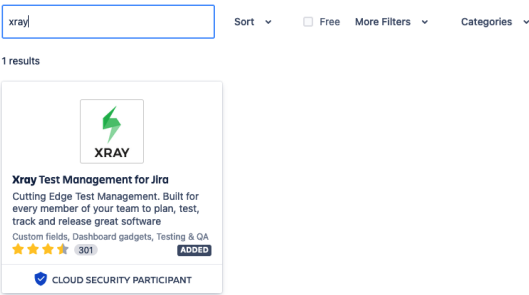


More information:

- [User guide](#)
- [Quick start guide](#)

In order to install Xray you just have to search the app and install it through the Apps entry of Jira.

Discover apps and integrations for Jira

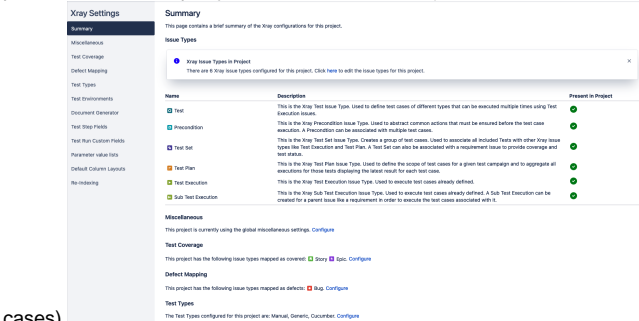


Please check [this page](#) for further details on the installation process.

Once installed, we then need to configure each project individually to enable certain features of Xray, such as the ability to create Tests for certain issues covered by those Tests (see [Quick Setup](#)).

The team needs to choose whether they want to have Tests alongside the same project where they have their requirements/stories or have it separated. Having a all-in-one scenario, where testing entities are stored in the same project where Story, Task and Bug issues are, is a good practice; however, Xray allows different approaches for project organization to suit what is best for your team (more info on [Project Organization](#)).

Aiming to provide flexibility to the teams, Xray is adaptable to match your way of work giving you configuration options to shape the tool to your team needs or you just can use the default (that will serve most



cases).

In Xray all the main entities are in fact Jira issues, this leverages all the advantages of being a Jira issue, such as permission control, workflow implementation, custom fields, screens customization, assignment, changes tracking, collaboration and much more...



More information:

- [Installation](#)
- [Configuration](#)
- [Organization](#)

What can you find more in Xray:

- Xray allows different [configurations](#) per project enabling the teams to adjust the product to their needs
- Xray issue types are handled as Jira issue types leveraging all the [advantages of Jira](#) regarding their handling.
- Xray configuration allows specific adaptations to the teams use case that you can find in [Quick Guides](#)

Testing activities

Let's go over the basic workflow teams use in order to enhance the quality of their products.

Usually teams go through a set of activities that will lead them to deliver a new version of their project/product. Let's look at those phases in more details.



- **Specification**
 - We specify Tests to validate requirements, or any other coverable issue. Optionally, they can be complemented with Preconditions, as means to specify the initial conditions to be assured before Tests can be run.
- **Organization**
 - Regarding Organization, we will cover the way that Tests are organized in the context of some project that will be key for the next phase.
- **Planning**
 - Test that are to be validated at a certain time for a certain target of the SUT (e.g., version, sprint) can be tracked through the Test Plans
- **Execution**
 - Once all the previous phase are concluded we can step into the execution of the Test that will provide valuable information over the quality of the release, executions will register all the information that is meaningful to enable decision taking, examples of information are the origin of the execution, the start/end/duration time, status, owner, environment, version, etc
- **Reporting**
 - After the testing activities have produced results we need ways to analyze what happened in a macro way and be able to have meaningful data to take decisions such as: "Can we deploy to production?", Reporting covers this area by producing reports that will help on the decisions we need to take on the project/product.

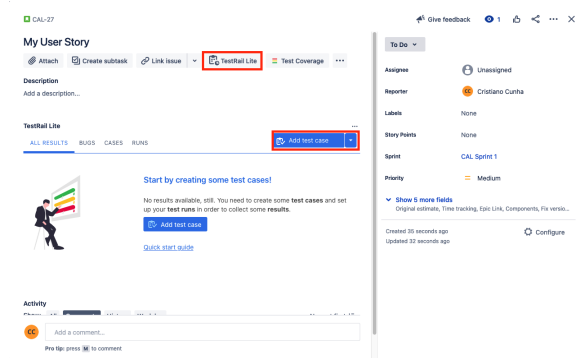
Specification

On all teams we have a moment dedicated to the specification of Epic/User Stories/Tasks/Tests where the team gathers and specifies/commits to an amount of work for a given time span and how to validate those. In our case we will focus on the activity of test case creation.

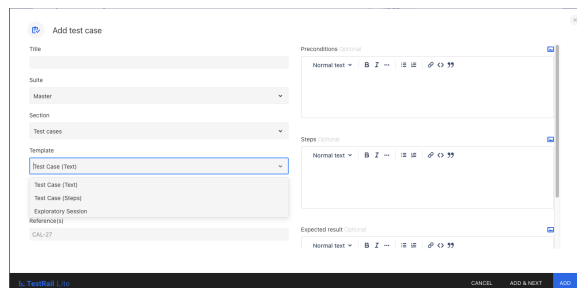
In order to investigate, evaluate and ascertain the completeness and quality of a project, developers perform quality assurance tests. Testing can be performed as individual components/units using Tests, or as a whole using Test Sets. The goal is to check and verify if a specific target requirement is applied, even when determined preconditions are set. It also helps prevent and uncover situations and defects that could negatively impact the overall usability of the final product.

TestRail Lite

Once TestRail Lite is installed into your Jira Cloud projects, you can create test cases inside your Jira issues.



A new panel opens up and allow you to define information to some fields and define the template that should be used to create the Test Case:



The template choice will influence the information to be filled in the right pane, the available templates are described below:

- **Test Case (Text)**
 - This template lets you define optional preconditions, steps, and expected results for a general test case in a flexible way.
- **Test Case (Steps)**
 - With this template you can determine specific steps your team should take when testing this case, and what the expected results should be. You will find a number next to each step.
- **Exploratory session**
 - Use this template if you need to test in a more exploratory way for your project defined by a certain goal or mission instead of steps and expected results.

You also have the possibility to define a Precondition for the Test Case, it is available as text field within the Test Case.

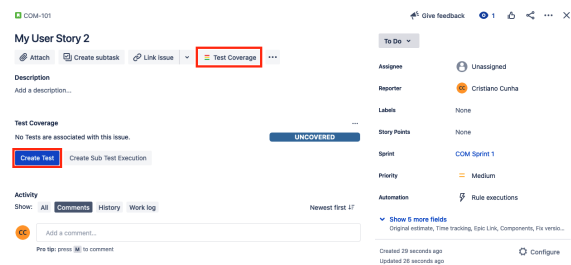
Xray

A full test **specification** may be done using just a [Test](#) issue, where you specify the steps and the expected results. Sometimes, you may prefer to abstract the initial conditions instead of writing them directly as the first Test steps, enabling you to use them in different tests. In this case, you may create one or more [Precondition](#) issues and associate them to a Test.

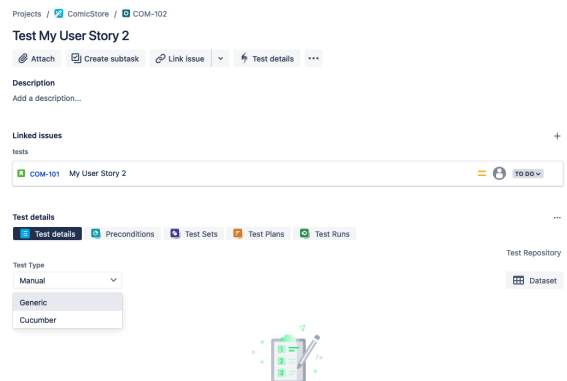
Although tests may be edited anytime, the immutability of tests is guaranteed whenever tests are scheduled to be executed in some test Execution.

You have several possibilities to create a Test, as all the Xray issues are Jira issues, you can use any create option available in Jira to create issues.

The following example will show one of those options, in this case the creation of an issue within the User Story in the Xray area:



Once you have added a new Test you can edit it to add detailed information, such as the Test Type to use:



The Test Type definition will define what are the information needed to each type:

- **Manual** (that will match the Test Case (Steps/Text) from TestRail Lite)
 - Xray provides a section to view, create and edit the **Test Steps**. These steps can be configured to have different fields; check this page [Test Step Fields](#) configuration page in the project settings.
- **Generic** (Usually used for automation but is also used to abstract an exploratory test charter and gather evidences for an exploratory session)
 - A Generic Test is an automated agnostic Test that is executed by an external tool. Execution results will then be imported back to Jira. An example might be a JUnit test that is used for quality assurance.
- **Cucumber** (Not matched in TestRail Lite)
 - Cucumber is a tool that executes plain-text functional descriptions as automated tests. The language that Cucumber understands is called **Gherkin**, which is a business-readable, domain-specific language that describes the software's behavior without detailing



More info:

- [Create Test Cases](#)

how that behavior is implemented. It is a line-oriented language and uses indentation to define structure. Line endings terminate statements (e.g., steps). Either spaces or tabs may be used for indentation (but spaces are more portable). Most lines start with a keyword.

These are the predefined Test Types made available in Xray but you have the possibility to create your own as **Xray enables you to create custom Test Types** that best match your needs.

You can also define a PreCondition and associate it to a Test as any Jira issue by linking it to the Test, as **Preconditions are a Jira issue** they **can be reused and shared amongst tests**.

Xray has an [Exploratory App](#) tailored for exploratory testing that integrates with Jira and Xray.

The Xray Exploratory App does not perform exploratory testing for you; however, it can surely help you stay focused while testing and ease documentation/note-taking so you can review all you did and discuss it later on with your team.



More info:

- [Writing Tests](#)
- [Manual Tests](#)
- [Generic Tests](#)
- [Cucumber Tests](#)
- [Parameterized Tests](#)
- [Preconditions](#)
- [Exploratory App](#)
- [Global settings: Test Types](#)

What can you find more in Xray:

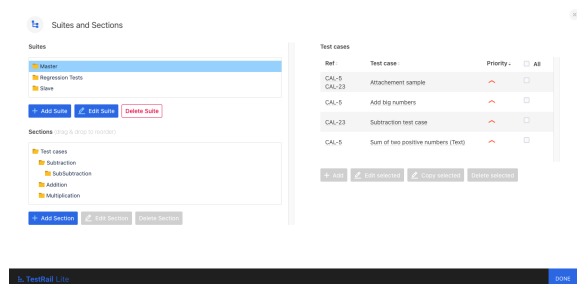
- [Test import/export](#)
- [API available to import Test Cases](#)

Organization

After having created all the tests the team require to do a proper validation (being properly linked to the requirement/story it covers) we need ways to enable the team to organize all of those tests.

TestRail Lite

TestRail Lite lets you organize your test cases by grouping them in Suites and Sections. This helps you find and reuse cases so that you can add them to test runs easily. By default TestRail Lite comes with a default suite created called "Master".



In this screen you can add Suites and Sections (with Sub-Sections) and associate the Test Cases to each. This panel also enable you to bulk edit Test Cases.

Xray

Organizing your tests is a crucial part of your daily testing activities. Since your project may have hundreds or thousands of tests, it is imperative that you organize them efficiently so you can find relevant tests quickly.

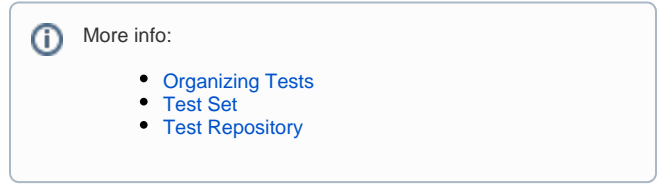
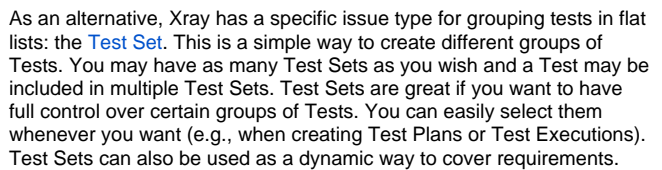
While many organize using labels, components, priority, etc., they are usually insufficient and would not scale according to your needs.

Xray provides two different but very effective ways of organizing Tests:

1. using Test Sets issues
2. using the Test Repository

The better match in Xray for TestRail Lite' Suites and Sections is the Test Repository.

The [Test Repository](#) provides a tree-like organization at the project level. Users can hierarchically organize Tests within folders and sub-folders. This folder concept is common in some tools and resembles user's own files/documentation organization in their computer's operating system.



What can you find more in Xray:

- [Tips for organizing Tests](#)

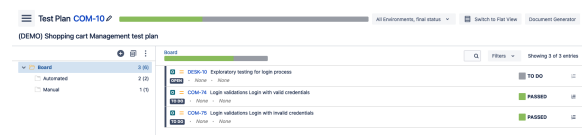
Planning is the activity where you decide your **testing strategy**, e.g., which covered issues you want to validate, how would you validate them, will it be manual or automated, how will the resources be allocated, when and who will execute the tests. Depending on several factors, you may want to prioritize some of the tests, or focus on specific requirements.

TestRail Lite does not have any entity to assist on the planning of Test Cases. You may achieve it with Test Suites and Sections even though they're not specifically tailored for that.

Although you may plan tests in [different ways](#), we recommend that you use the [Test Plan](#) issue specifically for this purpose.

In your Test Plan, you can put the Tests you want to track for a given version and the Test Plan will show you the consolidated results for those Tests. In other words, it will present the latest status of each Test, independent of the number of testing iterations (i.e., Test Execution) you make with them.

You may create a Test Plan per sprint, or for the whole release, for example; you decide.



Test	Environment	Status	Execution
COM-10	Exploratory testing for login process	Not Executed	0/0
COM-10	Exploratory testing for login process	Not Executed	0/0
COM-10	Exploratory testing for login process	Not Executed	0/0
COM-10	Exploratory testing for login process	Not Executed	0/0
COM-10	Exploratory testing for login process	Not Executed	0/0



More info:

- [Planning Tests](#)
- [Test Plan](#)
- [Test Plans in the Testing Board](#)

What can you find more in Xray:

- [Tips for planning tests](#)

Execution

A test execution is the activity that occurs between developing test scripts and reporting and analyzing said test results. This activity is more complex than starting a test and monitoring it to ensure that the test is running as expected. Actually, it involves

- Validating the test environment to ensure that it truly represents the production environment. If it is different from the environment designed for the tests to be run against, there is a high probability that tests might not work or will provide misleading data. For the same reason, tests should also be validated to ensure that they reflect the simulation design.
- Running the test based on the plans and test documents once it is determined that the test target is in an appropriate state to have performance tests run against it. After completing the test execution, a brief summary addressing what happened during the test should be compiled and added to the test log for future reference. Also, if the analysis of results leads to new priorities, then it is important to modify the tests, test designs, and subsequent strategies.
- Generating the test results and report any errors found to the development team.

TestRail Lite

After you have added a few test cases, you can start executing tests with TestRail Lite by adding a Test Run.

Open a Jira Issue and click on the arrow next to 'Add test case'. Select 'Add test run' from the dropdown menu and a new window will pop up.

After adding test runs in your Jira Issues, you can find them below the **TEST RUNS** tab.

Xray

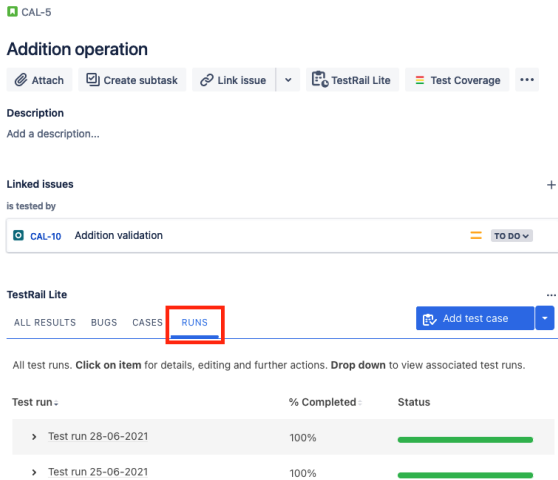
Xray provides the [Test Execution](#) issue (that will match the TestRail Lite Test Run), which acts as an assignable task for executing a group of tests (either manual or automated) and collects the corresponding test results for a specific project version and environment.

A [Sub-Test Execution](#) is similar to a Test Execution, but created as a sub-task of a requirement.

Normally, you will create Test Executions from Test Plans, or create ad hoc Test Executions (directly from a Test).

Depending on the strategy used to perform the validation of your product/project you will be able to create Test Executions from different origins:

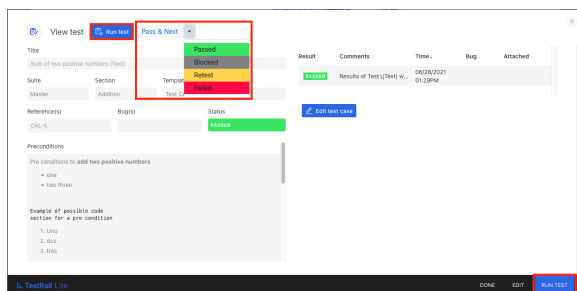
- Entering in the Test details and checking the *Test Runs* section



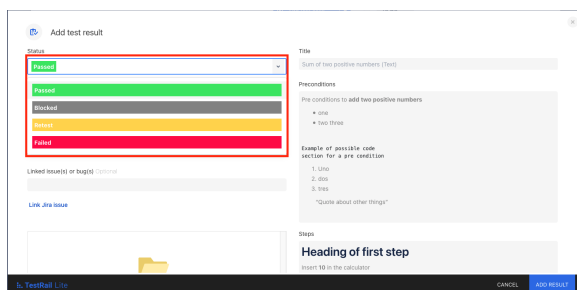
Select one of the test runs from the list to run the test cases.

Once you are in the details of the Test Run you can pass/fail/block/retest the Test automatically if you use the drop down in the top menu.

If you want to add comments, screenshots or bugs to the Test Run you must choose *Run Test* option from the top menu or bottom right buttons.



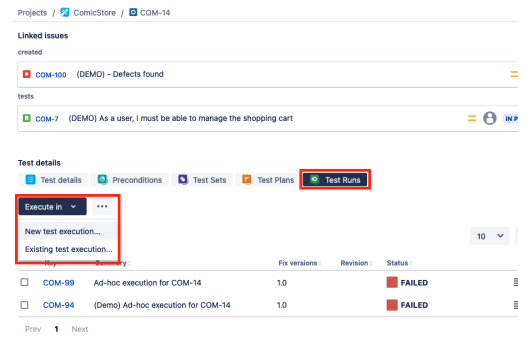
If you choose to *RUN TEST* you will taken to an execution panel where you can add comments, screenshots, link bugs or issues, see the details of the Test Case and define its overall status.



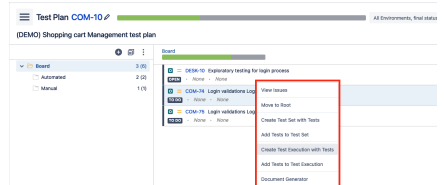
What will change amongst the different templates is the right panel that will show different details depending on the test template used.

More Info:

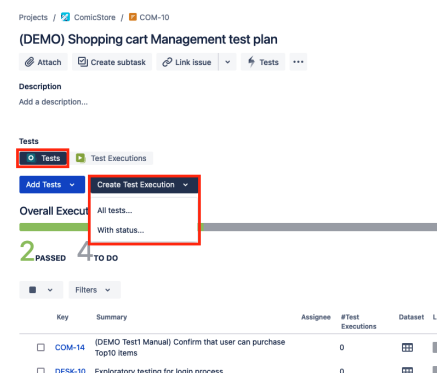
- [Create Test Runs](#)
- [Test Runs](#)
- [Test Results Areas](#)



- If using Test Plan as a strategy, you will be able to check the Test Execution status:
 - In the Test Plan view

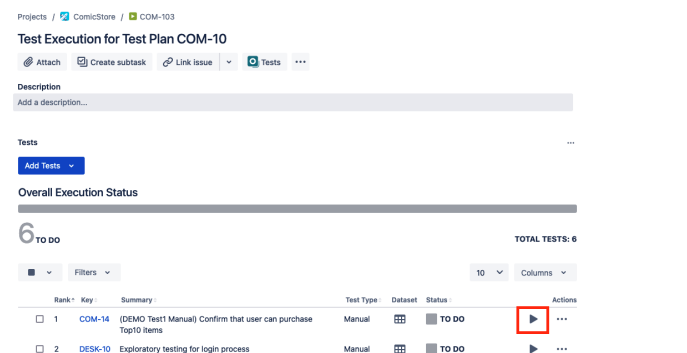


- In the Test Plan details within the Test section

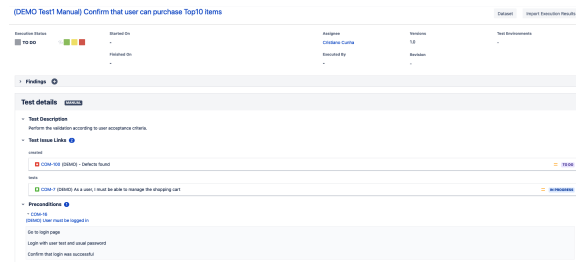


Once you have your Test Executions created, you're ready to *execute* them. Each time you schedule a given Test for execution, a *Test Run* will be created and Xray will keep the information related to that specific execution of that specific Test.

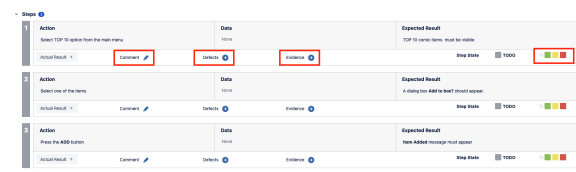
On the detail view of the Test Execution you will able to start the execution using the *Play icon* present next to each Test:



That option will take you to the execution panel where you will have all information needed to perform the execution (Description, link to requirements and bugs, Preconditions)



and for each Step you will be able to add a comment, create a defects, associate an evidence and define the status of that step



These possibility to add details per steps will enabled a more specific report for a defects being able to pin point where it failed and what were the steps performed so far.



More info:

- [Executing Tests](#)
- [Test Execution](#)
- [Sub-Test Execution](#)
- [Execute Tests](#)
- [Test Runs](#)
- [Working with Test Environments](#)
- [Automated Tests \(Import/Export\)](#)

What can you find more in Xray:

- Xray support [importation of Test Results](#) in different formats through our API
 - Xray [import results](#)
 - Xray [Integrates with several Testing Frameworks](#)
- Xray have available a set of [different Integrations](#) (CI/CD Tools and others)
 - [Understanding coverage](#)

Reporting

In order to obtain relevant metrics on the progress of your tests as well as the completeness of your requirements, Xray provides several ways that make it easier for the entire project team to have a clear view of the overall readiness of the project.

As part of your daily tasks, you may use dedicated reports and gadgets in order to evaluate your testing progress.

TestRail Lite

TestRail Lite only has available the list of Test Runs that you can export as an HTML or PDF document.

You can reach this report through the left menu entry:

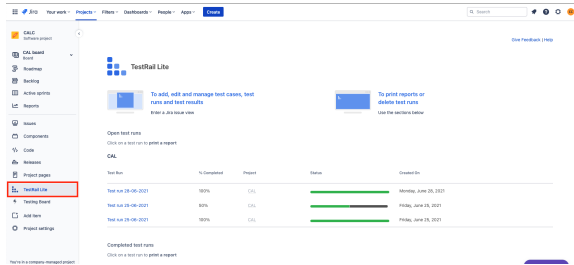
Xray

Xray essentially uses Jira issues, so you can always use JQL to obtain relevant results. Xray also provides some [custom fields](#) and some entity properties that may be queryable using JQL.

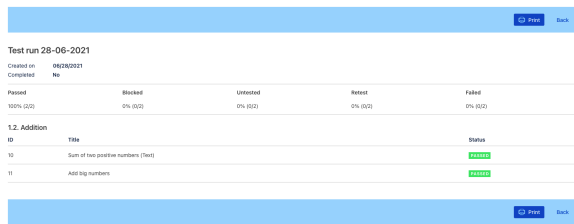
The custom fields may be exposed directly in the standard Jira search result table or in tools, such as Jira's **Filter Results**.

In summary, your reporting may be accomplished by:

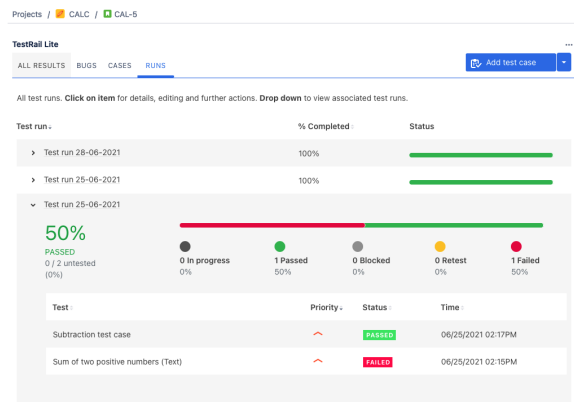
- using standard [JQL queries](#)
- consulting the specific panels of Tests/Test Sets/Test Plans /Test Executions
- using the [Built-in Reports](#)



Once you access one of the Test Runs you will be taken to the following page where you can print the report:



The other way is through the requirement where you have defined the Test into:



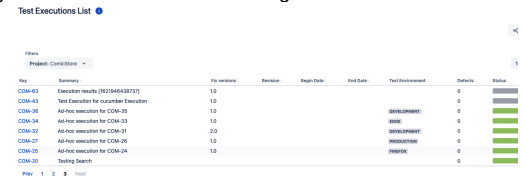
In this last option you will not have the option to print the report.



- building out dashboards using [gadgets](#)

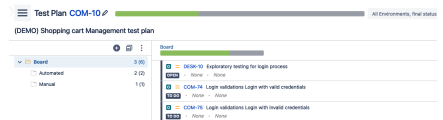
You will be able to access the Test Executions results from the following sources:

- Entering in the Test details and checking the *Test Runs* section

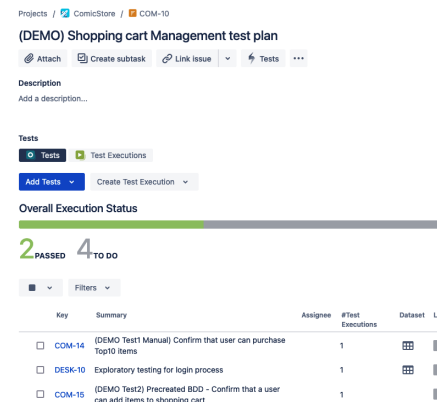


- If using Test Plan as a strategy, you will be able to check the Test Execution status:

- In the Test Plan view



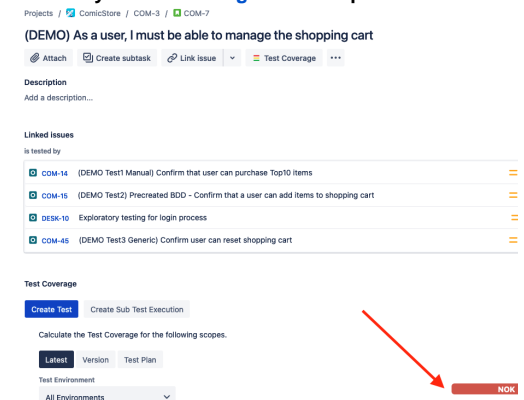
- In the Test Plan details within the Test Execution section



- Accessing the report [Test Executions List](#) (and with full details [Test Runs List](#))



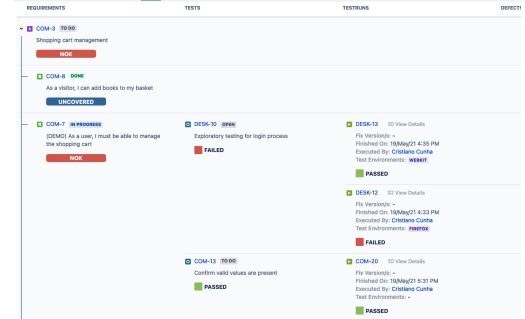
- The more important one is on the requirement itself where we will have the visibility of the coverage (calculating the status with a combination of all status of all Tests associated to it), **bringing the visibility of the coverage to the requirement level.**



Let's take a look on two major Reports available through Xray:

- Traceability

- This report enables you to follow the life of a requirement in both forwards and backwards direction (i.e., from its origin all the way through tests, test runs and defects). It facilitates analysis of the overall requirements coverage status.



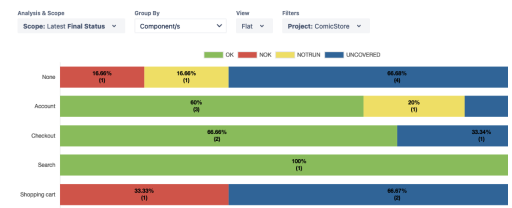
- Coverage

- In order to analyze the current status of your release, or even other releases from the past, you need to evaluate the status of requirements, taking into account the respective results.

You might want to analyze just the requirements implemented in a given version or you may want to analyze requirements from earlier versions based on your regression testing.

The overall requirements coverage report provides the means to analyze your requirements statuses in real-time.

Test Coverage Report



More info:

- [Tracking, Measuring and Analyzing](#)
- [Reports & Analysis](#)
- [JQL Queries](#)
- [Built-in Reports](#)
- [Gadgets](#)
- [Customize Reports with Xporter](#)

What can you find more in Xray:

- Xray [Document generator](#)
- Xray have a set of different [Built-in Reports](#)

Moving forward

Xray is a complete solution for test management, so there's much to explore.

We recommend attending one of our [webinars](#) for a more in-depth of Xray, or watch a previous recording of a [overall demo of Xray](#).