Execute Tests

Overview

You can either execute your tests manually or automatically. If it is an automated test, you can still manually set the test status for that execution. You can also create custom statuses.

Tests are always executed within a Test Execution context, even if it's an ad hoc execution. You can execute tests inline (i.e., define the test run status), directly in the Test Execution issue screen, or you can use the Execution Screen, which is detailed extensively below.

A Test Execution validates if all the Tests in a specific environment target passed. In order to execute, monitor the progress, and report the results, each Test has its own Execution page details. Each Test associated with a Test Execution is called a **Test Run**. A Test Run is like an instance of the Test that contains a copy of the test definition (e.g., description, steps) and the result of that test in the context of that Test Execution.

When a Test definition is modified, the respective Test Runs aren't automatically updated. This is because the execution of a Test depends directly on its definition. Each time a Test Run is created, the Test definition is copied to it. This guarantees that when the Test definition changes, the Test Runs are not affected and will always contain the original (i.e., when it was added to the Test Execution) Test definition.

You are allowed to execute a Test if:

- you have permission to **"Browse"** the Test Execution issue
- you have permission to **"Resolve"** the Test Execution issue
- the Test Execution issue is not in a status configured to disallow executions
- the Test issue is not in a status configured to disallow executions

---

ⓘ **Please note**

Find more about how you can disallow executions based on the workflow status of Test or Test Execution issues at:

- Global Settings: Miscellaneous
- Using Jira workflows for testing purposes

Even though you might not have permission to execute a Test Run and if the execution has already started, you can see the execution details page in **read-only** mode. An informative message will be shown to at the top of the execution page if the current Test Run is in read-only mode.

When you execute a Test, the Xray Execution Screen is shown. This screen is accessible from the Test Execution issue view page and the Test issue view page (Test Runs section), and this is where the execution results are displayed and registered. This page contains information about the Test definition and the Test Execution issue context to allow you to execute the Test without exiting this page.

The execution screen is divide in 5 sections. The first one is **Navigation and Actions bar**, then the **Execution Status and General Information**, after that the **Findings** section, the **Test Details** and **Activity** sections:

- Navigation and Actions bar
- Execution Status and General Information
- Findings
    - Defects
    - Evidence
    - Custom Fields
    - Comment
- Test Details
    - Test Description
    - Test Issue Attachments
    - Test Issue Links
    - Parameters
    - Preconditions
    - Steps/Definition/Scenario or Scenario Outline
    - Iterations
    - Result contexts
- Activity

Book Store / Test Plan: STORE-2 / Test Execution: STORE-5 / Test: STORE-1   Test 1 of 1

Test visitors can unsubscribe from the book store Newsletter   ⓘ   Dataset   Import Execution Results

Execution Status	Timer	Started On	Assignee	Versions	Test Environments
EXECUTING	⏸ 00:00:52 ↺	15/Feb/2022 04:45 PM	Rafael Morais	-	-
	No time logged	**Finished On** -	**Executed By** Rafael Morais	**Revision** -	

> **Findings** ⊕

Test details   MANUAL

∨ **Test Issue Links** ①

tests

   ▢ STORE-4 As a visitor, I can register to the book store Newsletter   ≡ TO DO

∨ **Preconditions** ①

   > STORE-3 - **The user is subscribed to the Newsletter**

∨ **Steps** ③

1	Action	Data		Expected Result
	Navigate to the newsletter page from the landing page	None		The newsletter must be visible.
	Actual Result ⌄ Comment ✎ Defects ⊕ Evidence ⊕		Step State	PASSED

2	Action	Data		Expected Result
	Press **Unsubscribe** to the newsletter	None		A confirm dialog appears prompting the user for conformation.
	Actual Result ⌄ Comment ✎ Defects ⊕ Evidence ⊕		Step State	PASSED

3	Action	Data		Expected Result
	Press **Confirm** to unsubscribe	None		User must receive feedback that he was unsubscribed successfully.
	Actual Result ⌄ Comment ✎ Defects ⊕ Evidence ⊕		Step State	EXECUTING

> **Activity**

Navigation and Actions bar

This bar is always available to allow you to navigate the Test issue hierarchy. It also contains a toolbar with the **Return to Test Execution** button that allows you to jump to the current Test Execution issue page. The **Next** and **Previous** buttons allow you to quickly navigate to the execution screen of the other Tests in the same Test Execution, if applicable.

On this bar, you can also find actions to **Export test definitions** and **Import execution results**. This is useful if you need to do an ad-hoc execution for an automated Test. Usually, automated tests are executed in a Continuous Integrating platform and the results are imported automatically into Xray (creating new Test Execution issues for the results) but you can also execute each automated Test individually and import the execution result.

Test visitors can unsubscribe from the book store Newsletter

Test 1 of 3

Dataset   Import Execution Results

Export to Cucumber

If you're executing a Cucumber Test, an option to export the Test as a feature file will appear.

Import Execution Results

If you already have the test results, just use this action to import the report files into Xray and update a specific Test Run.

Dataset

If you want to edit a test run dataset, you can do it by clicking on the **Dataset** button. To learn more about it click on Parameterized Tests.

Exploratory App

If your test is an unstructured test, that option will appear and if clicked the Exploratory app will open. To learn more about it click Exploratory App.

Execution Status and General Information

The **Execution Status** describes the current Test progress. The possible (native) status for a Test Run are **TODO**, **EXECUTING**, **FAIL,** and **PASS**. You can also create custom statuses in the Xray Test Statuses Settings global page.

You can change the status of the execution directly through this field. It supports inline editing and once activated, it will display a set of statuses that you can set the Test Run. If the option for setting the Test Run status manually without having to execute all Test Steps (or Examples, as in the case of Cucumber Tests) is enabled in the Xray administration page, then you can set the Test Run Status to any status, independent of the current aggregate status of the Test Steps (or Examples). If this option is disabled, then the status that you can set through this field will be limited to the available transitions based on the current Steps or Examples aggregated statuses.

The status section contains other relevant fields such as:

- **Timer** - The time elapsed since the last time the timer started and the total time logged into this execution
- **Assignee** - the User assigned to perform the current test execution
- **Executed By** - the last User that changed the status of the current test run
- **Versions** - the target release version tested by the current test execution
- **Revision** - the source code and documentation version used in the current test execution
- **Started On** - the date and time the execution of the current test started
- **Finished On** - the date and time the execution of the current test finished

Execution Status	Timer	Started On	Assignee	Versions	Test Environments
EXECUTING	00:31:43	15/Feb/2022 03:11 PM	Rafael Morais	1.0	-
	No time logged	Finished On	Executed By	Revision	
		-	Rafael Morais	-	

You can change the Test Run assignee directly on the field. It supports inline editing and once activated, you can filter by user and then select the one you want to assign to the Test Run.

If the dates are configured to be edited, you can edit the **Started On** date by clicking the calendar icon and choosing the date and time. You can check your configurations by going to Jira Administration > Add-ons > Miscellaneous.

Timer

Timer

| **❚❚** | 00:00:16 | ↺ |

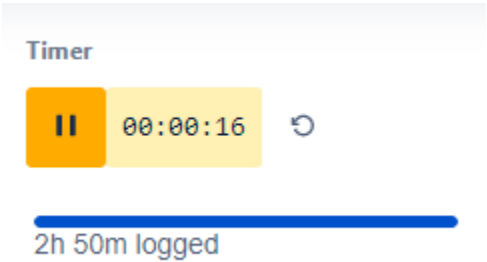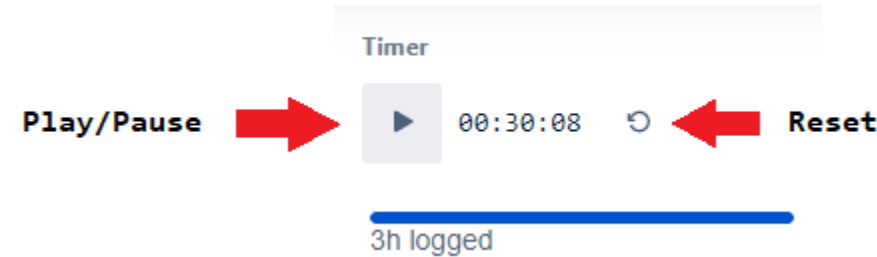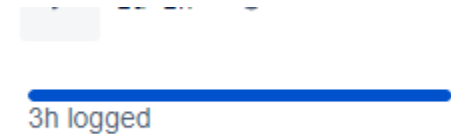2h 50m logged

The timer is used to help the user track the time spent in the execution of the test.
This component will only be visible if the corresponding setting is active in the project settings.
The timer itself can be started, paused and reset manually using the button to do so:

Timer

Play/Pause ➡ ▶ 00:30:08 ↺ ⬅ **Reset**

3h logged

You can also edit the value on the timer by clicking on it if the **Set the value of the Time Tracker** setting is active in the project settings.

The timer will also automatically start when the status of the Test Run changes to **Executing** and it will pause when it changes to a final status (i.e. **Passed**, **Failed**), it will also reset when it the Test Run changes to **Todo**.
Below the timer there is a component that tracks the time logged for this execution:

3h logged

By clicking on it, you can log more time into it:

Time added in this dialog will be added to the work log of the respective Test Execution:

Findings

This section is where the Defects, Evidences, Comments and Custom Fields added are displayed.

Defects

This field lists the overall and step defects related to the current execution. It is possible to **Create new Defect** issues (or sub-tasks) or **Add existing Defects** by clicking the respective option.

Group Defects

"Show Steps Defects" is an option that you can see in Manual tests, that allow you to see overall defects and defects associated with steps in the same panel.

This feature is optional and you can enabled/disabled it on the respective panel section.

Example:
When the toggle is on, both defects appear in the panel and a label (Global/Step/Iteration + Step) is shown at the start of the defect line to easily distinguish each of them. When turned off, only the overall defects are shown.

Show Steps Defects
Toggle On

Evidence

This field lists all the attachments related to the current execution. Clicking **Add Evidence** opens up the **Attach Files** dialog box. Check here for more information about attachments.

Group Evidence

"Show Steps Evidence" is an option that you can see in Manual tests, that allows you to see overall evidence and evidence associated with steps in the same panel.

This feature is optional and you can enable/disable it on the respective panel section.

Example:
When the toggle is on, all evidence appears in the panel and a label (Global/Step/Iteration + Step) is shown at the start of the evidence line to easily distinguish each of them. When turned off, only the overall evidence are shown.

Custom Fields

You define additional custom fields for Test Runs. These fields can be useful to add extra information to Test Runs, usually only available when or after executing Tests.

Test Run custom fields can be configured by project and by Test Type. Therefore, these settings will not affect other projects within your Jira instance. For instance, it is possible to have custom fields just for Manual Tests within a project.

If there are any custom fields defined for the scope of the Test Run, they will appear on the "**Custom Fields**" section.

It is also possible to define custom fields that are **mandatory**. If this is the case, users will not be able to set a final status on the Test Run if there is any required custom field that is not populated.

Comment

This field is inline editable so you can input comments about the current execution without reloading the page.

It supports the Jira markup wiki language.

Test Details

This section includes some fields present on the Test issue. It often contains useful information for testers to execute the Test.

Test Description

This section includes the current Test issue description. It often contains useful information for testers to execute the Test.

This field supports the Jira markup wiki language.

Test Issue Attachments

This section shows all attachments associated with the Test.

Test Issue Links

This section shows all links associated with the Test.

Test Issue Links 1

tests

☐ STORE-7 As a visitor, I can register to the book store Newsletter ↑ **TO DO**

Parameters

This field shows the parameters' values defined in the dataset when there is only a single row in it. To learn more about it click on Parameterized Tests.

Parameters 3

${Landing Page Link} = www.1234.com ${Unsubscribe_link} = Unsubscribe ${Confirmation_Button} = Confirm

Preconditions

This field shows the Preconditions issues associated with the current Test.

Preconditions 1

> STORE-40 - The user is subscribed to the Newsletter

Steps/Definition/Scenario or Scenario Outline

Within that section, you will find the Test script to execute. Depending on the Test Type, this section can display **Steps** (for manual Tests), **Definition** (for unstructured Tests), and **Scenario** or **Scenario Outline** (for Gherkin scenarios).

Manual Test Step Results Table

A **Manual Test** execution screen contains the **Test Steps**, with the Action, attachments and all the other configured fields to be used as a reference while testing. The Test Steps field configuration used is the one defined in the Test Run's Test project.

By clicking on each step, the following fields become editable:

- the **Actual Result** field which contains the actual result of the execution of the step;
    - This field can be edited using the WYSIWYG markdown editor.
    - It's possible to refer to an Evidence previously added to any Step or Global execution
- the **Comment** field, which contains any notes about the corresponding step;
- the **Defects** field, which contains any defect associated with the corresponding step. It is also possible to input new defects by clicking on:
    - **Create Defect** to create a new Defect
    - **Create Sub-Defect** to create a new Sub-Defect (sub-task)
    - **Add Defects** to add an already existing defect
- the **Evidences** field, which contains any attachment associated with the corresponding step. It is also possible to input new attachments by clicking on the **+** button, which opens the **Attach Files** dialogue box
- the **Status,** which contains the current step testing status. This can be changed by clicking on any of the statuses in front of your Step Status.

Note: Changing the Status on each Step will affect the **Execution Status** field on the Execution Screen.

- **Source:** "Backoffice"

Show More

| Actual Result ⌄ | Comment 🖊 | Defects ➕ | Evidence ➕ | Step State | PASSED |

| **10** | **Execution Domain** | **Action** | **Data** |
| | POSTMAN | Execute the following request. | https://staging-server.local:8682/store/order/<ORDER_ID> |

Expected Result

The following response must be returned:

```
1  {
2      "orderId": <ORDER_ID>,
3      "status": "PROCESSING",
4      "source": "Backoffice",
5      "items": [
6          ...
7      ]
8  }
```

Actual Result

| Body | Cookies | Headers (10) | Test Results | | Status: 401 Unauthorized | Time: 121ms | Size: 408 B |

| Pretty | Raw | Preview | Visualize | JSON ▾ |

```
1  {
2      "error": "Could not find authentication data on request"
3  }
```

| Actual Result ⌄ | Comment 🖊 | Defects ➕ | Evidence ➕ 👁 (2) | Step State | FAILED |

Gherkin Tests

Gherkin Tests always display the Test scenario on the execution page. Usually, these tests are automated and their result can be imported into Xray. However, it is also possible to execute these Tests manually by setting the overall result of the Test Run, or by setting the result of each example (in case of scenario outlines).

Test Details CUCUMBER SCENARIO OUTLINE

⌄ Scenario Outline

```
1  Given I have entered <input> into the calculator
2  And I have entered <input_2>
3  When I press <button>
4  Then the result should be <output>
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     | 1 | 40 | add | 41 ||
```

Examples

A **Scenario Outline** contains a table of examples that define several inputs and outputs for the given scenario. Each example is, therefore, a set of values that will be replaced on the scenario.

The examples section details these examples, containing an overall status bar in the header and a table with all the examples below. Each example can have the status: PASS, FAIL, EXECUTING or TODO.

A scenario outline test is PASS only if all examples passed.

A scenario outline test is FAIL if at least one of the examples failed.

A scenario outline test is TODO if at least one of the examples did not run and all the other examples passed or todo.

<input_1>	<input_2>	<button>	<output>	Duration	Status
20	30	add	50	-	PASSED
2	5	add	7	-	PASSED
0	40	add	40	-	FAILED
1	40	add	41	-	TODO

Unstructured Tests

Unstructured Tests only contain the **definition** field on the Test Details. This field can specify a test script, an ID to an external (automated) test case, or an Exploratory Test charter. Similar to Gherkin Tests, most unstructured test results will be imported or set by another tool. However, it is also possible to set the status of unstructured tests manually.

Test Details GENERIC

˅ Definition

net.thucydides.showcase.junit.features.search.SearchByKeywordTest.Search for articles by shop name

Iterations

When this is a test run of a data-driven test, this field shows all the iterations corresponding to the rows of the dataset closest to the test run. For each iteration, the preconditions and steps appear with the expected parameter values for it. To learn more about it click on Parameterized Tests.

All operations for test steps are also available within an iteration, namely adding defects, evidence, and the actual result.

This section also features a progress bar displaying the status of all iterations.

Filtering iterations

It is also possible to filter the iterations by status. A filter button is provided next to the iterations progress bar. Clicking on this button will open a panel with all the available statuses. Choose one or more statuses and press Apply.

Status calculation

When a step status is changed, the iteration status will be updated automatically, according to the step statuses and, in turn, the overall test status will also be updated based on all iterations statuses.

Test Details MANUAL

˅ Iterations 18

> **Iteration 1 -** In Search of Lost Time $34 5 Marcel Proust Yes New Paperback Yes 1   PASSED

> **Iteration 2 -** One Hundred Years of Solitude $20 4.9 Gabriel Garcia Marquez Yes Used Paperback Yes 1   PASSED

> **Iteration 3 -** The Great Gatsby $39 4.7 F. Scott Fitzgerald No New Kindle Yes 1   PASSED

> **Iteration 4 -** In Search of Lost Time $34 5 Marcel Proust Yes New Paperback No 1   PASSED

> **Iteration 5 -** One Hundred Years of Solitude $20 4.9 Gabriel Garcia Marquez Yes Used Paperback No 1   PASSED

> **Iteration 6 -** The Great Gatsby $39 4.7 F. Scott Fitzgerald No New Kindle No 1   PASSED

˅ **Iteration 7 -** In Search of Lost Time $34 5 Marcel Proust Yes New Paperback Yes 2   EXECUTING

  ˅ **Steps** 12

1	**Action**	**Data**	**Expected Result**
	Add 2 books of In Search of Lost Time into the basket.	None	After items are added, a confirmation message appears mentioning 2 items of In Search of Lost Time were added to the basket.
	Actual Result ⊻ Comment 🖉 Defects ⊕ Evidence ⊕		**Step State** PASSED

2	**Action**	**Data**	**Expected Result**
	Click on the basket icon located on the top right toolbar of the app.	None	The basket page is displayed containing all 5 items.
	Actual Result ⊻ Comment 🖉 Defects ⊕ Evidence ⊕		**Step State** PASSED

3	**Action** | Attachments 🖇 (1)	**Data**	**Expected Result**
	Press the **Checkout** button to start the checkout process.	None	The checkout process is initiated asking the user the address details.

Automated Results

Xray supports the import of multiple execution results for the same Test Run. Multiple executions for the same Test Run specifies different results for different contexts/environments. Multiple executions are only supported for Automated Tests. Manual Tests do not have this concept, yet.

When a Test Run contains multiple executions imported from external sources like Calabash and Xamarin Test Cloud, these executions are presented in a "**Results**" section on the execution page.

This section contains a Table with all executions for the same Test Run. The table has the following columns:

- **Context:** An execution context name (in the case of Xamarin Test Cloud results, the mobile device, and operating system version)
- **Error Message:** A possible error message if the result has failed
- **Duration:** The complete duration of an execution result
- **Status:** The execution results status

Results ③

Context				Duration	Status
⌄ CHROME				119 millisec	PASSED
<input_1>	**<input_2>**	**<button>**	**<output>**	**Duration**	**Status**
› 8	4	divide	2	119 millisec	PASSED
⌄ 12	3	divide	4	1 millisec	PASSED
Hooks					
around FactorialCalculator.around()			⊙(1)	0 millisec	PASSED
before FactorialCalculator.setUp()			⊙(2)	0 millisec	PASSED
after FactorialCalculator.tearDown()			⊙(2)	0 millisec	FAILED
FAILED HERE					
after CalculatorPressOne.tearDown()				0 millisec	PASSED
Backgrounds					
Given a HP calculator			⊙(5)	0 millisec	PASSED
Given a calculator I just turned on			⊙(5)	0 millisec	PASSED
Steps					
Given I have entered 4 into the calculator				0 millisec	PASSED
When I press factorial				0 millisec	PASSED
Then the result should be 24 on the screen				0 millisec	PASSED
› 3	1	divide	3	19 secs	FAILED
⌄ Internet Explorer				119 millisec	FAILED
<input_1>	**<input_2>**	**<button>**	**<output>**	**Duration**	**Status**
› 8	4	divide	2	119 millisec	PASSED
› 12	3	divide	4	1 millisec	PASSED
› 3	1	divide	3	19 secs	FAILED

Activity

The Activity field lists all the activities related to the current execution.

andre.rodrigues changed the comment of step 3.
New Value
Still executing this step
Today 1:18 PM

For data-driven tests the iteration number will appear in the activity list:

⌄ Activity

MC Mélanie Castro **ITERATION 1** changed status of step 1 from TODO to PASSED
Today 02:05 PM

Attaching Screenshots

To attach a screenshot, paste the image from your clipboard using the relevant keyboard shortcut. (For Mac OSX, use **CMD+V;** for Windows, use **Ctrl+V.**)

Subtract two numbers

Import Execution Results   ⌃ Return to Test Execution   ‹ Previous   Next ›

Execution Status   **EXECUTING**

Assignee:
Bruno Conde

Versions: **1.0**
Revision: -

Started On: 29/May/2018 11:23 AM

Executed By:
André Miguel Pereira Rodrigues

Test Environments: Android

Attach Screenshot

File name *

screenshot-1

Location

Overall Evidences ⌄

Save   Cancel

Comment

Click to add comment

Defect ⌃

Execution Evidences (0)   Add Evidences ⌃

▶ **Execution Details**

Test Description   ⌃

This is the description for this Test

Test Issue Links (2)

| created | 🔴 | CALC-15 Error while subtracting 2 numbers | ↑ | TO DO |
| tests | 🟢 | CALC-11 Addition Operation | ↑ | TO DO |

...ments (1)   ⌃

📄 calculator.png

395.5 kB 25/May/2018 2:02 PM

Preconditions (1)   ⌃

⌄ CALC-14
Turn on the Calculator

Enter a file name for the screenshot you are attaching (defaults to `screenshot-` concatenated with the next number available) and choose where you want to place the evidence (Overall Execution or Test Steps).

Attach Screenshot

Execution Defects (0)   Add Defects   Create Defect ⌃

Step #1 Evidences

Step #2 Evidences

Step #3 Evidences

Overall Evidences ⌄

Save   Cancel

Click **Save** to attach the captured image to your Test Run.

Updating the definition of the Test after the execution started

When a Test definition has changed, a new dialog box will appear at the top of the Execution page informing you that the current execution is obsolete. You will be prompted with two buttons: **Reset** and **Merge**.

⚠ **Test definition has been modified.**

You can reset the execution of this test by pressing the "Reset" button. This will load the new test definition and delete the current execution data.
If you wish to update the Test definition and maintain the current execution, you can click on the "Merge" button.

Reset · Merge

Reset Execution

You can choose to reset the Execution or keep it for historic record. If you reset the execution, all information previously saved (status, evidence, defects) will be erased.

Merge Execution

You can choose to merge the new Test definition into the current status of execution. This will keep steps that weren't changed along with the associated results/comments/evidence. It will also load new steps (moved, erased, or created), preconditions, or attachments definitions. This will always force a recalculation of Test Run Status. For instance, if the current Test Run Status is FAIL because of a step and this step is removed from Test, if you merge, then the Test Run Status will change to whatever the value combination of the others steps returns.