Gherkin (BDD Tests)

- Overview
  - More about Cucumber
- Defining Gherkin Tests
  - Using the Issue dialog
  - Inline Editing Xray Tests
- Automated Steps Library
  - Key concepts
  - Steps
  - Learn more
- Export to Cucumber

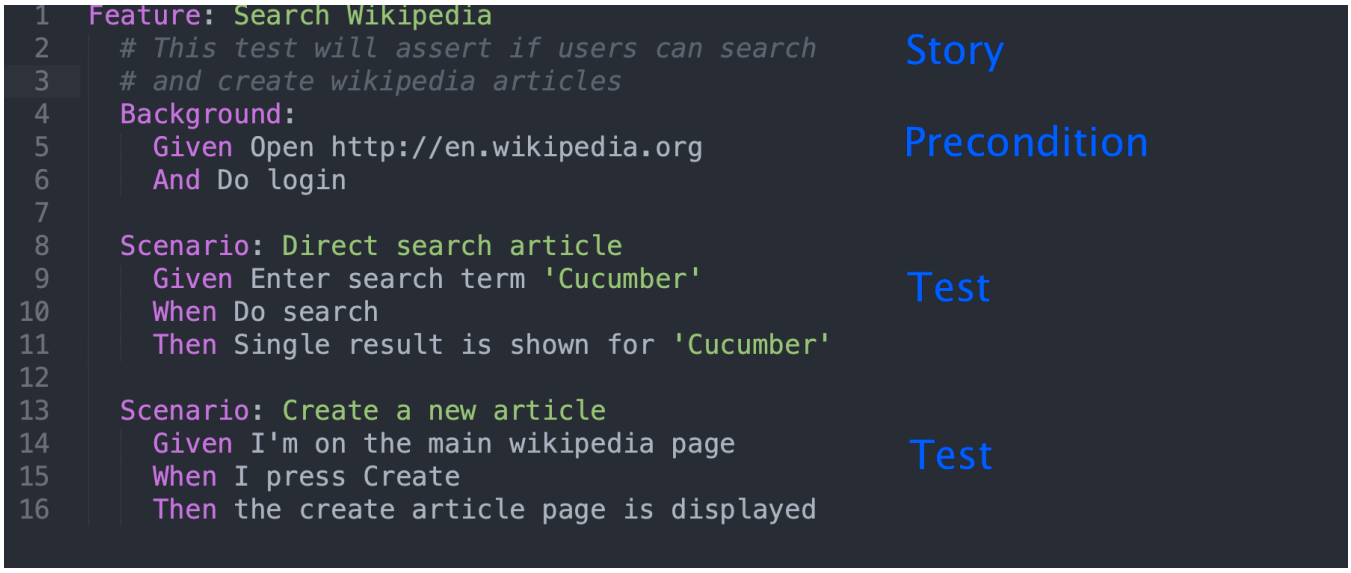
Overview

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 how that behavior is implemented. Gherkin is a line-oriented language that uses indentation to define the 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. Gherkin has become the go-to language of choice for Behaviour-Driven Development or BDD. Gherkin is now supported by many other tools other than Cucumber.

For more information on Cucumber and the Gherkin language, please consult the Wiki and Tutorials at http://cucumber.io

More about Cucumber

Cucumber tests are defined in **.feature** files. Each feature file normally contains all test scenarios for a given software feature. The image below is a simple Cucumber feature file and shows how the different sections are mapped to Jira/Xray entities.



Each **Feature** is mapped to a Requirement in Jira (e.g. Story, Epic, or any other issue type that you consider a Requirement).

The **Background** clause is mapped to a **Precondition** issue in Xray.

Each **Scenario** or **Scenario Outline**, is mapped to a **Test** issue in Xray.

Cucumber Tests (or scenarios) are of two **Scenario Types**:

- **Scenario:** a set of instructions exclusively composed by constant values, which, when executed by a tool, will only give a static result (e.g., 2 + 3 = 5)
- **Scenario Outline:** a generic set of instructions containing variables, which, when executed by a tool, will prompt varied results according to the specified input.

In Xray, you can specify the **scenario** and the **scenario type** directly on the Test issue.

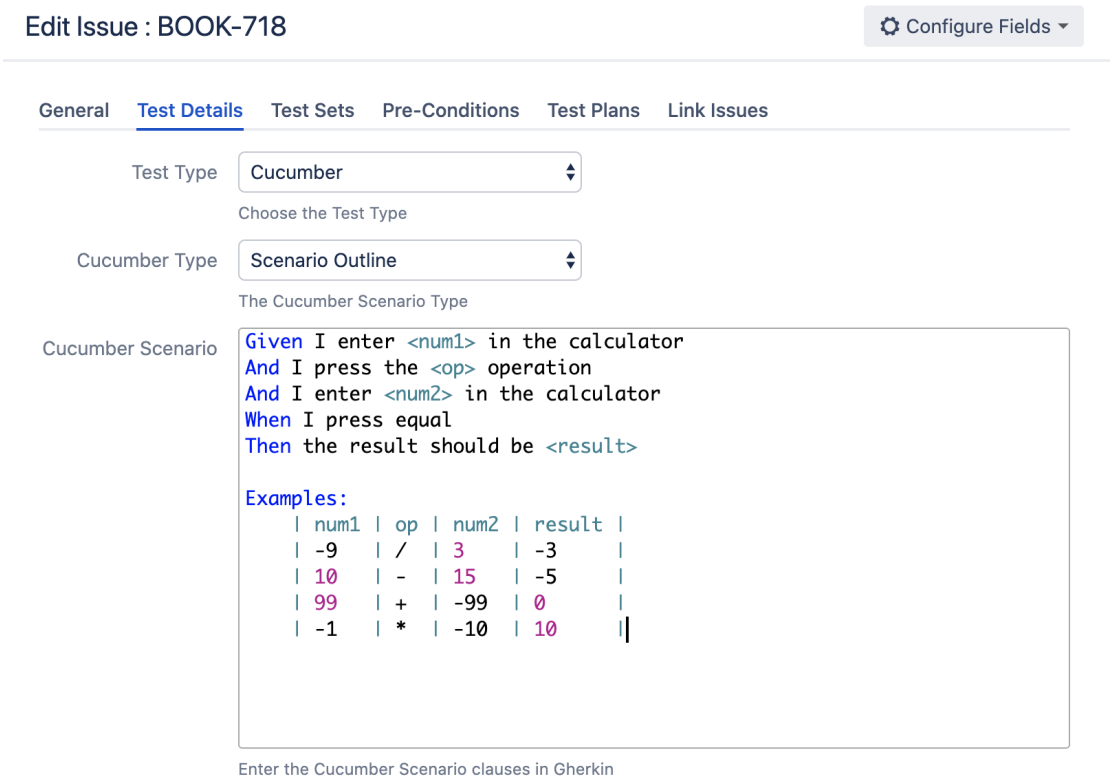
- Overview
  - More about Cucumber
- Defining Gherkin Tests
  - Using the Issue dialog
  - Inline Editing Xray Tests
- Automated Steps Library
  - Key concepts
  - Steps
  - Learn more
- Export to Cucumber

Defining Gherkin Tests

In Xray, you can specify the **scenario** and the **scenario type** directly on the Test issue.

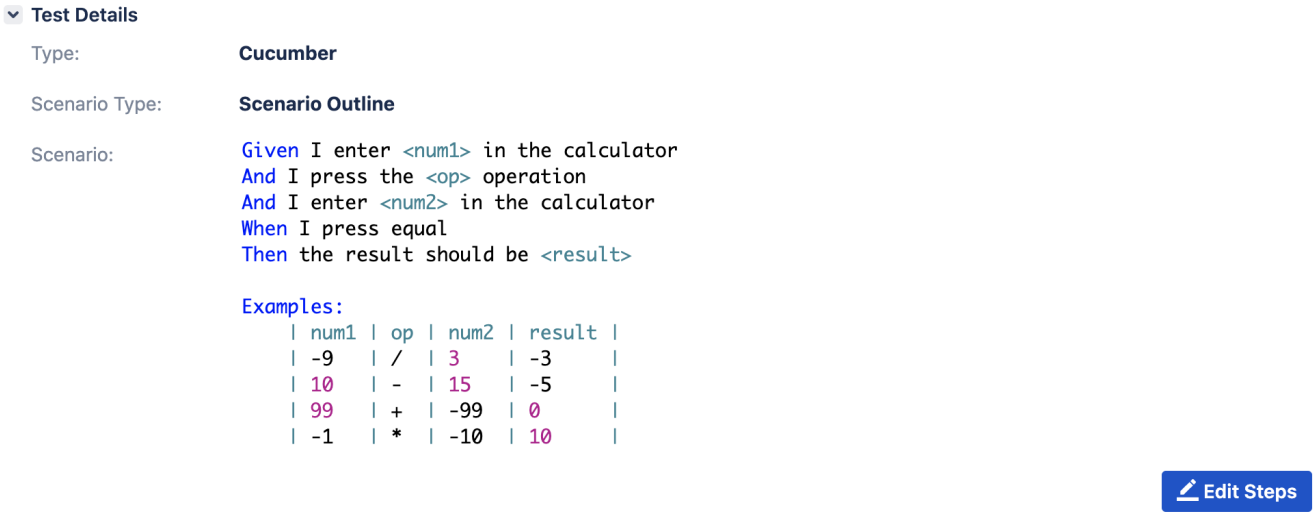
Using the Issue dialog

When creating or editing Gherkin Tests, you can set the **scenario** directly on the Jira create/edit dialog or page. Just go into the Test Details tab, and choose a Gherkin Test Type. Now you can set the scenario and scenario type:

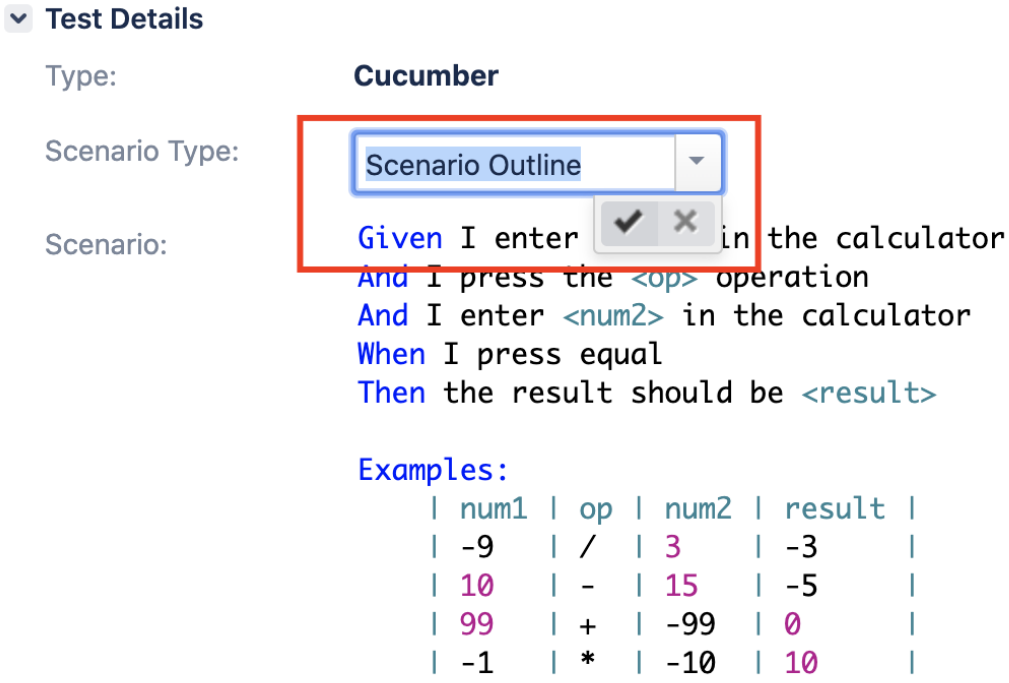


Inline Editing Xray Tests

When viewing Gherkin Test issues in Jira, you can also use inline editing capabilities to set the **scenario** and **scenario type** fields. This prevents full page reloads when modifying these fields.



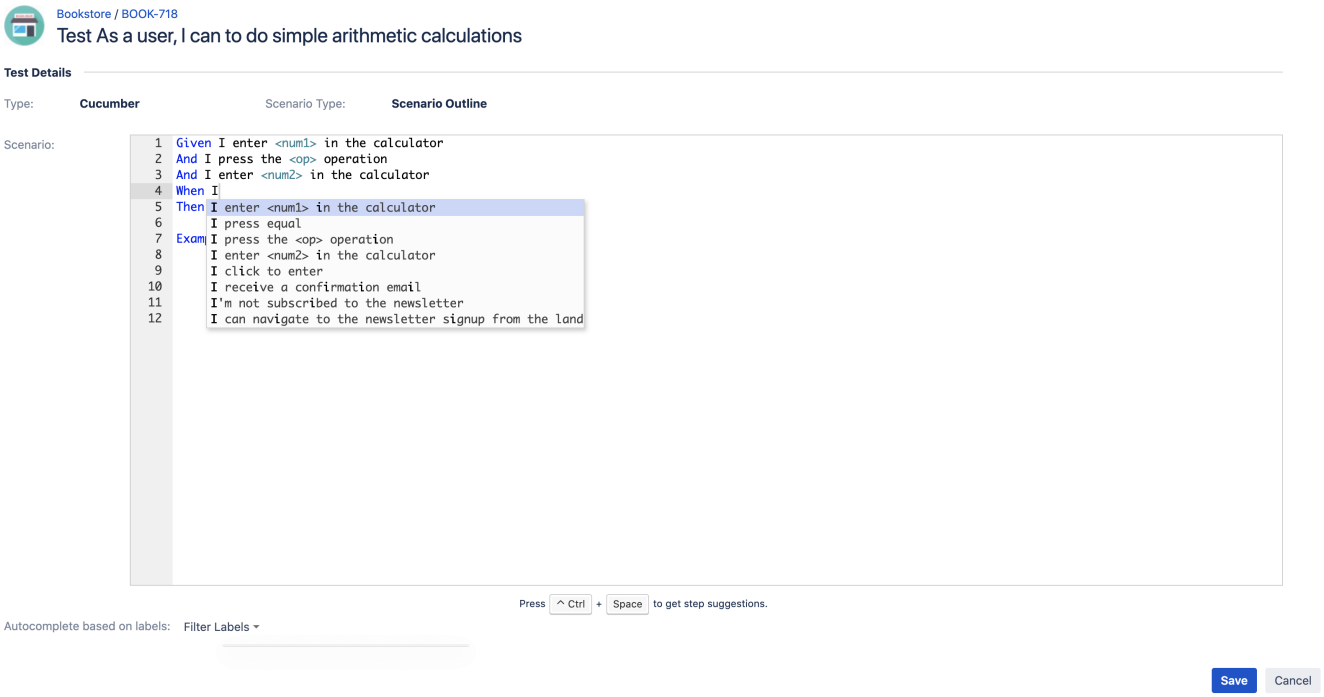
In order to edit the **scenario type** you just need to hover over the field and click on it. The field will display an edit mode where you can change the value using a single select list field. After choosing the value, just press the check button near the field to save the new value.



To edit the **scenario** itself, you need to use the **Edit Steps** button.

This button will open a dedicated dialog where you can edit the scenario steps using **syntax highlight** and also take full advantage of the **Automated Steps Library** described in the next section.

When done editing the steps, just press **Save** to persist the changes or press **Cancel** to discard them.



You can only edit the **scenario** and **scenario type** fields if you have **permission** to edit the Test issue.

Automated Steps Library

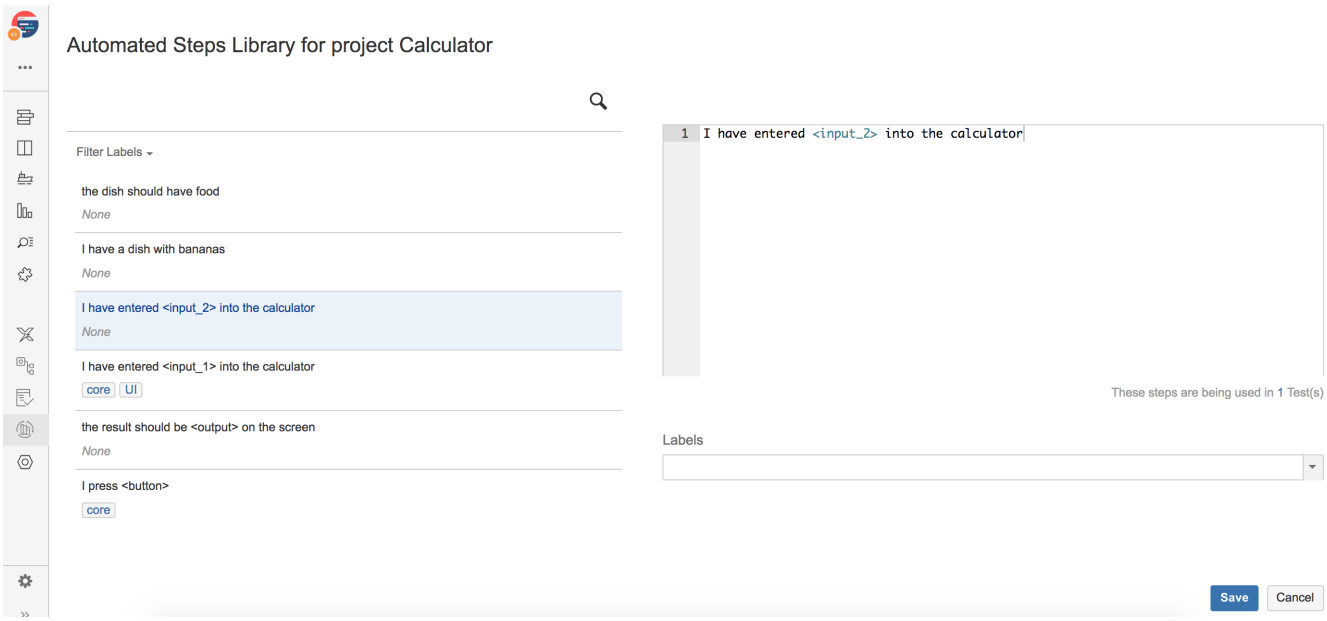
The **Automated Steps Library** provides a centralized place to manage automated steps and allows for easy refactoring of steps (Cucumber/Gherkin only).

Refactoring is the keyword and is essential to maintain your automation code base sane.

Since steps are persisted and indexed, they can then be used for autocomplete purposes during the creation/update of Cucumber Scenarios/Backgrounds, facilitating their creation, and avoiding duplicated or messy steps.

A general setting, available in Global Preferences, allows you to globally opt-in or opt-out for this feature. If the Automated Steps Library is not enabled, users can still specify Cucumber Scenarios/Backgrounds, although lacking the autocomplete ability.

Key concepts



The **Automated Steps Library** provides a per-project organization for all the steps used in automated (Cucumber only) Tests and Pre-Conditions belonging to that project.

In other words, the library provides a centralized point to manage all Gherkin sentences (i.e. the Cucumber "steps") of all the Tests and Pre-Conditions of a given project.

Steps are automatically added to the **Automated Steps Library** of a project whenever a Cucumber Test/Pre-Condition of that project is created or updated. Likewise, they're automatically removed if they are not being used by any of those entities, in that project.

Steps are always, and indirectly, automatically created in the library whenever creating/editing a Cucumber Test/Pre-Condition. Steps cannot be created within the **Automated Steps Library** itself.

On the left side of UI, steps can be searched and filtered by labels, while on the right side of the step, content can be edited and saved.

The **Automated Steps Library** provides the means to:

- refactor steps, by easily renaming them (i.e. changing their specification)
- refactor steps, by merging similar steps and avoiding duplicated specifications & associated code
- see which Tests/Pre-Conditions are using a particular step
- label steps, so they can be categorized, in order to ease their management and the step autocompletion during Test/Pre-Condition edition

In the **Automated Steps Library** you cannot:

- create/remove steps (this is done automatically, whenever creating/editing/removing Cucumber Tests/Pre-Conditions)

Steps

Steps, from the current Automated Steps Library perspective, correspond to Gherkin sentences used by Cucumber Tests/Pre-Conditions (i.e. Cucumber Scenario, Scenario Outline, Background).

Steps are case insensitive and their content is trimmed (i.e. spaces removed at the start/end).

Gherkin reserved keywords (e.g. *Given*, *When*, *Then*, *And*) are not stored in the library, so these two following steps are equally stored as "I have a calculator":

Step as written in a Cucumber Test/Pre-Condition	Stored step
`Given I have a calculator` `And I have a calculator`	`I have a calculator`

Steps can be updated/refactored in the Automated Steps Library; they can also be implicitly updated or created whenever editing or creating a Cucumber Test/Pre-Condition.

However, they cannot be created from within the Automated Steps Library UI, in order to avoid dummy/unused steps. In other words, if a step appears in the Automated Steps Library then most probably it is being used by some entity. Exceptions may arise though, especially if the library is disabled somewhere in time or if for some reason Xray was unable to process an update/creation Jira event.

Integrity Checker, or a specific option in the project settings, may be used to force a per-project synchronization of steps, in case you find orphaned or missing steps in the library.

Learn more

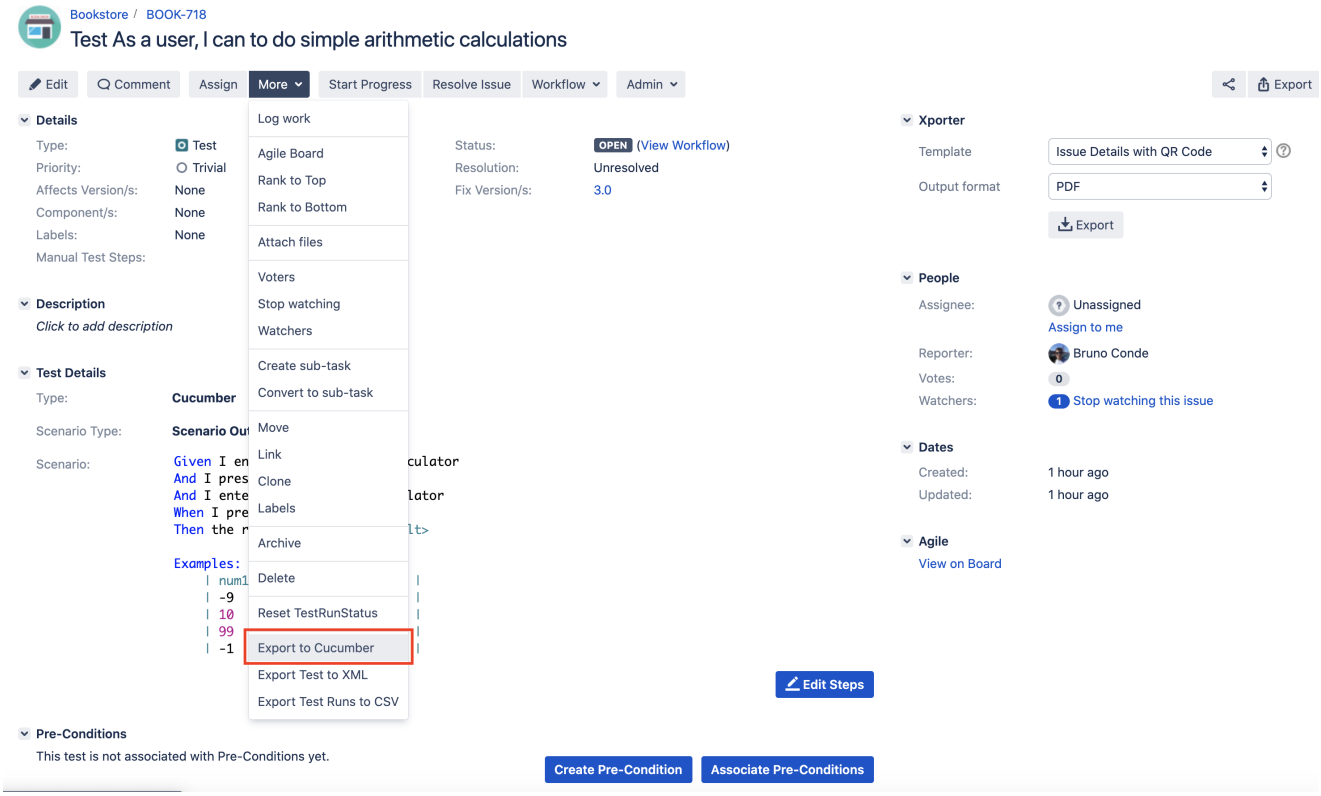
For a fully comprehensive list of the available operations and how to perform them, please check the Automated Steps Library page.

Export to Cucumber

Gherkin Tests in Xray can be exported to **.feature** files ready to be executed in Cucumber. The generated feature files follow the rules defined in Export Cucumber Features and take into account the context of the Jira Test issue.

A single Cucumber **feature** file should be generated and prompted for download, unless the target Test is associated with multiple **Requirements**, which, in that case, will prompt a **FeatureBundle** zip file containing a single Cucumber feature file for every Requirement associated with the target Test.

In order to generate a Cucumber feature(s) for a given Test issue:

Step 1: Open the Cucumber Test issue view page.

Step 2: Select **More > Export to Cucumber.** The file will be downloaded to the browser.