# Testing using Cucumber in Java

## Overview

In this tutorial, we will create some tests in Cucumber for Java.

The test (specification) is initially created in Jira as a Cucumber Test and afterwards, it is exported using the UI or the REST API.

## Requirements

- Install Java
- Add the dependency of cucumber-jvm (i.e. cucumber-java) to your maven "pom.xml" file
- Clone of the Github repository "cucumber-java-skeleton"

## Description

We will use the code from the Github repository "cucumber-java-skeleton", with slight changes in order to make the dummy Test pass.

The first step is to create a Cucumber Test, of Cucumber Type "Scenario", in Jira. The specification would be exactly the same as the one provided in the original repository.

After creating the Test in Jira and associating it with requirements, etc., you can export the specification of the test to a Cucumber .feature file via the REST API or the **Export to Cucumber** UI action from within the Test Execution issue.

The created file will be similar to the original, but will contain the references to the Test issue key and the covered requirement issue key.

**new feature after export /cucumber-java-skeleton/src/test/resources/skeleton/belly.feature**

```
@ABC-100
Feature: Belly

 @ABC-122
 Scenario: a few cukes
   Given I have 42 cukes in my belly
   When I wait 1 hour
   Then my belly should growl
```

You can change the implementation of the steps in order to make them pass quickly.

**/cucumber-java-skeleton/src/test/java/skeleton/Stepdefs.java**

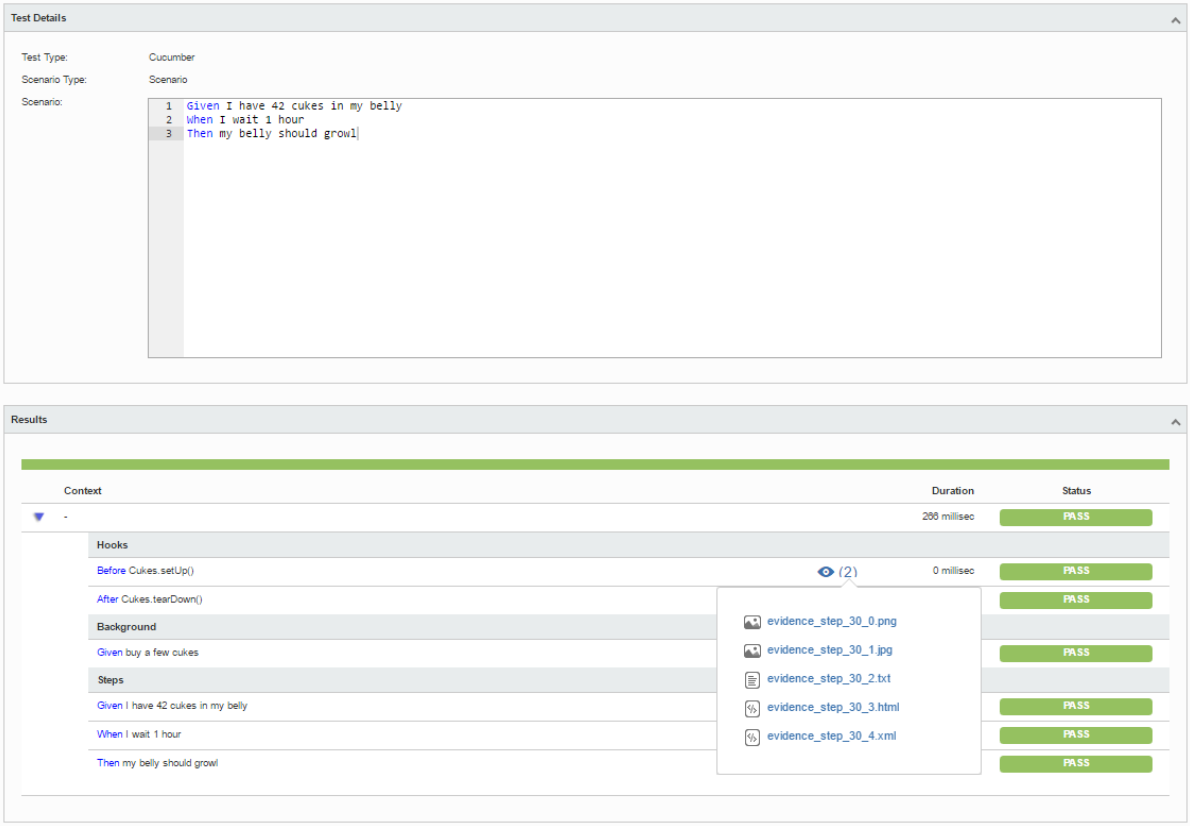
```
package skeleton;
import cucumber.api.java.en.Given;
import cucumber.api.java.en.Then;
import cucumber.api.java.en.When;

public class Stepdefs {
    @Given("^I have (\\d+) cukes in my belly$")
    public void I_have_cukes_in_my_belly(int cukes) throws Throwable {
        Belly belly = new Belly();
        belly.eat(cukes);
    }


    @When("^I wait (\\d+) hour$")
    public void I_wait_hours(int hours) throws Throwable {
        Thread.sleep(hours*0);
    }

    @Then("^my belly should growl$")
    public boolean my_belly_should_growl() throws Throwable {
        return true;
    }
}
```

After running the tests and generating the Cucumber JSON  report (e.g., data.json), it can be imported to Xray via the REST API or the **Import Execution Results** action within the Test Execution.

```
mvn compile test -Dcucumber.options="-p json:data.json"
```

The execution screen details will provide information on the test run result.

The Cucumber Scenarios Example/Result details (i.e., **Hooks, Backgrounds** and **Steps**) are only available for executions done in Xray v2.2.0 and above.




The icon (2) represents the evidences ("embeddings") for each **Hook, Background** and **Steps**, but is only available for executions done in Xray v2.3.0 and above.

**Learn more**

Please see Testing in BDD with Gherkin based frameworks (e.g. Cucumber) for an overview on how to use Cucumber Tests with Xray.

# References

- https://docs.cucumber.io/installation/java/
- https://github.com/cucumber/cucumber-java-skeleton
- Automated Tests (Import/Export)
- Exporting Cucumber Tests - REST