Testing using Cucumber in Ruby/JRuby

Overview

In this tutorial, we will create some tests in Cucumber for Ruby (or JRuby).

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 Ruby or JRuby
- Install the "cucumber" gem

Description

After creating a Cucumber Test, of Cucumber Type "Scenario Outline", in Jira, 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 following:

1_CALC-889.feature

```
@REQ_CALC-889
Feature: As a user, I can calculate the sum of 2 numbers
```

```
@TEST_CALC-908 @UI @core
Scenario Outline: Cucumber Test As a user, I can calculate the sum of 2 numbers
                                           Given I have entered <input_1> into the calculator
                                           And I have entered <input_2> into the calculator
                                           When I press <button>
                                           Then the result should be <output> on the screen
                                             Examples:
                                               | input_1 | input_2 | button | output |
                                               20
                                                    | 30 | add | 50
                                                                                 | 2
                                                        | 5
                                                                 add
                                                                           7
                                                                         1
                                                       40
                                               0
                                                                add
                                                                         40
                                                        50
                                               | 4
                                                                 add
                                                                         54
```

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.

cucumber -x -f json -o data.json

0



The execution screen details will not only provide information on the test run result, but also of each of the examples provided in the Scenario Outline.

The Cucumber Scenarios Example/Result details (i.e., Hooks, Backgrounds and Steps) are only available for executions done in Xray v2.2.0 0 and above.

rest De	etaiis						
Test T	Type:	Cucumber					
Scena	ario Type:	Scenario Outline					
Scena	ario:	1 Given I have entered (1 2 And I have entered (1) 3 The stered (1) 4 Then the result should 5 Examples: 7 Lingut_1 input_2 8 120 130 9 12 15 10 1 4 50	nput_l> into the calculator nt_l> into the calculator be coutput> on the screen button output add 58 add 7 add 54 add 54				
Exampl	les						
Exampl	les <input_1></input_1>	<input_⊅< td=""><td> bu</td><td>utton> <outpu< td=""><td>ab</td><td>Duration</td><td>Status</td></outpu<></td></input_⊅<>	 bu	utton> <outpu< td=""><td>ab</td><td>Duration</td><td>Status</td></outpu<>	ab	Duration	Status
Exampl	<input_1></input_1>	<input_2> 30</input_2>	 bu add	utton> <outpu s 50</outpu 	Þ	Duration 128 millisec	Status PA SS
Exampl	<input_1> 20 Hooks</input_1>	≪nput_⊅ 30	 bu add	utton> <outpu i 50</outpu 	ib	Duration 128 millisec	Status PA SS
Exampl	<input_1> 20 Hooks Before feature</input_1>	≪input_≥ 30 resistep_definitionsicaloulator_steps.rb:7	⊲bu add	utton> <outpu d 50</outpu 	it-	Duration 128 millisec 0 millisec	Status PA SS PA SS
Exampl	<input_1> 20 Hooks Before feature After feature</input_1>	≪input_2> 30 resistep_definitions/calculator_steps.db:1 ssistep_definitions/calculator_steps.db:11	 bu add	utton> <outpu s 50</outpu 	1Þ	Duration 128 millisec 0 millisec 0 millisec	Status PAISS PAISS PAISS
Exampl	<input_1> 20 Hooks Before feature Background</input_1>	<input_2> 30 resistep_definitions/calculator_steps.rb:11 ssistep_definitions/calculator_steps.rb:11 d</input_2>	<bu data<="" td=""><td>utton> <outpu s 50</outpu </td><td>IP</td><td>Duration 128 millisec 0 millisec 0 millisec</td><td>Status PA SS PA SS PA SS</td></bu>	utton> <outpu s 50</outpu 	IP	Duration 128 millisec 0 millisec 0 millisec	Status PA SS PA SS PA SS
Exampl	<pre>les <input_1> 20 Hooks Before feature Background Given a calc</input_1></pre>	<input_2> 30 resistep_definitions/calculator_steps.rb.71 ssistep_definitions/calculator_steps.rb.11 d .vulator I just turned on</input_2>	 bu add	utton> <outpu s 50</outpu 	۹Þ	Duration 128 milisec 0 milisec 0 milisec 128 milisec	Status PA SS PA SS PA SS PA SS
T	les ≤input_1> 20 Hooks Before feature Background Given a calc Steps	<input_2> 30 res/step_definitions/calculator_steps.rb:7 ss/step_definitions/calculator_steps.rb:11 d ulator I just turned on</input_2>	 bu add	utton> ≺outpu s 50	it-	Duration 128 milisec 0 milisec 0 milisec 128 milisec	Status PASS PASS PASS PASS
Exampl	<pre><input_1> 20 Hooks Before feature Background Given a calc Steps Given 1 have</input_1></pre>	<pre><input_2> 30 ires/step_definitions/calculator_steps.nb.7 sistep_definitions/calculator_steps.nb.11 d usilator i just turned on e entered 20 into the calculator</input_2></pre>	<bu add</bu 	utton> <outpu i 50</outpu 	it-	Duration 128 millisec 0 millisec 0 millisec 128 millisec 0 millisec	Status PASS PASS PASS PASS PASS
Exampl	<input_1> 20 Hooks Before feature Background Given a calc Steps Given 1 have And 1 have e</input_1>	<pre><input_2> 30 ires/step_definitions/calculator_steps /b: 7 is/step_definitions/calculator_steps /b: 11 d vulator I just turned on entered 20 into the calculator entered 30 into the calculator</input_2></pre>	 du add	utton> <outpu ś 50</outpu 	tÞ	Duration 128 millisec 0 millisec 0 millisec 128 millisec 0 millisec 0 millisec	Status PASS PASS PASS PASS PASS PASS
T	<pre><input_1> 20 Hooks Before featur Background Given a calc Steps Given 1 have And 1 have e When 1 press</input_1></pre>	<pre><input_2> 30 ires/step_definitions/calculator_steps rb:7 is/step_definitions/calculator_steps rb:11 d vulator I just turned on entered 20 into the calculator entered 30 into the calculator s add</input_2></pre>	 du add	utton> <outpu d 50</outpu 	tÞ	Duration 128 milisec 0 milisec 0 milisec 128 milisec 0 milisec 0 milisec 0 milisec	Status PASS
T	<pre>iss input_t> 20 Hooks Before feature Background Given a calo Steps Given I have e When I press Then the ress </pre>	<pre><input_2> 30 30 res/step_definitions/calculator_steps r/b 7 siststep_definitions/calculator_steps r/b 11 d culator I just turned on e entered 20 into the calculator entered 20 into the calculator s add suit should be 50 on the screen</input_2></pre>	 add	utton> <outpu s 50</outpu 	¢ (2)	Duration 128 milises 0 milises 128 milises 128 milises 0 milises 0 milises 1 milises 1 milises	Status PASS
Exampl	les <input_t> 20 Hooks Before feature Background Given a calo Steps Given I have e When I press Then the res 2</input_t>	<pre><input_2> 30 resistep_definitions/calculator_steps .fb 7 sistep_definitions/calculator_steps .fb 11 d ulator I just turned on e entered 20 into the calculator entered 30 into the calculator s add sult should be 50 on the soreen 5</input_2></pre>	 du add	utton> <outpu s 50</outpu 	¢ (2)	Duration 128 millisec 0 millisec 0 millisec 128 millisec 0 millisec 0 millisec 0 millisec 0 millisec 0 millisec	Status PASS PASS
Exampl	<pre>seles </pre> 20 Hooks Before feature Background Given a calo Steps Given 1 have And 1 have e When 1 press Then the res 2 0	<pre><input_>> 30 rres/step_definitions/calculator_steps .fb:7 d ulutator I just turned on e entered 20 into the calculator es add uut should be 50 on the screen 5 40</input_></pre>	 add add add	utton> <outpu s 50</outpu 	₽ (2) (2)	Duration 128 milisee 0 milisee 120 milisee 0 milisee 0 milisee 0 milisee 1 milisee 1 milisee 0 milisee 0 milisee	Status PASS

0

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

1

- https://cucumber.io/docs/reference/ruby
 Automated Tests (Import/Export)
 Exporting Cucumber Tests REST