4. Cucumber

Cucumber Introduction

Cucumber is a tool based on Behavior Driven Development (BDD) framework which is used to write acceptance tests for web application. It allows automation of functional validation in easily readable and understandable format (like plain English) to Business Analysts, Developers, Testers, etc.

Cucumber feature files can serve as a good document for all.

Cucumber can be used along with Selenium, Watir, and Capybara etc. Cucumber supports many other languages like Perl, PHP, Python, .Net etc. In this tutorial we will concentrate on Cucumber with Java as a language.

Cucumber Basics:

In order to understand cucumber we need to know all the features of cucumber and its usage.

#1) Feature Files:

Feature files are essential part of cucumber which is used to write test automation steps or acceptance tests. This can be used as live document. The steps are the application specification. All the feature files ends with .feature extension.

Sample feature file:

```
Feature: Create Customer Feature
  As a user of XYZ Bank,
  I want to open an account
  So that I have to create an account firstly
  Scenario: Create the customer successfully
    Given I open the website http://www.globalsqa.com/angularJs-protractor/BankingProject/#/manager/addCust
    Then I verify that add customer screen appears
    When I type first name as Thuy , last name as Nguyen , postcode as 05550
    When I click submit button
    Then I check the customer first name as Thuy , last name as Nguyen , postcode as 05550 has been existed in the customer list

  Scenario: Create the customer successfully
    Given I open the website http://www.globalsqa.com/angularJs-protractor/BankingProject/#/manager/addCust
    Then I verify that add customer screen appears
    When I type first name as Hong , last name as Ho , postcode as 05147
    When I click submit button
    Then I check the customer first name as Hong , last name as Ho , postcode as 05147 has been existed in the customer list
```

#2) Feature:

This gives information about the high level business functionality (Refer to previous example) and the purpose of Application under test. Everybody should be able to understand the intent of feature file by reading the first Feature step. This part is basically kept brief.

#3) Scenario:

Basically a scenario represents a particular functionality which is under test. By seeing the scenario user should be able to understand the intent behind the scenario and what the test is all about. Each scenario should follow given, when and then format. This language is called as “gherkin”.

1. **Given:** As mentioned above, given specifies the pre-conditions. It is basically a known state.
2. **When**: This is used when some action is to be performed. As in above example we have seen when the user tries to log in using username and password, it becomes an action**.**
3. **Then:** The expected outcome or result should be placed here. For Instance: verify the login is successful, successful page navigation.
4. **Background:** Whenever any step is required to perform in each scenario then those steps needs to be placed in Background. For Instance: If user needs to clear database before each scenario then those steps can be put in background.
5. **And**: And is used to combine two or more same type of action.

Example:

Example of Background:

Background:

Given *user navigates to http://way2automation.com/way2auto_jquery/index.php*
And all the junk values are cleared

#4) Scenario Outline:

```
Feature: Create Customer Feature
  As a user of XYZ Bank,
  I want to open an account
  So that I have to create an account firstly
  Scenario: Create the customer successfully
    Given I open the website <website>
    Then I verify that add customer screen appears
    When I type first name as <firstName> , last name as <lastName> ,
postcode as <postcode>
    When I click submit button
    Then I check the customer first name as <firstName> , last name as
<lastName> , postcode as <postcode> has been existed in the customer
list

Examples:
| website                    | firstName  | lastName | postcode |
|
http://www.globalsqa.com/angularJs-protractor/BankingProject/#/manager/a
ddCust | Thuy       | Nguyen   | 05550    |
|
http://www.globalsqa.com/angularJs-protractor/BankingProject/#/manager/a
ddCust | Hong     | Ho    | 05147 |
```

Note:

1. As shown in above example column names are passed as parameter to When statement.
2. In place of Scenario, you have to use Scenario Outline.

3. Examples are used to pass different arguments in tabular format. Vertical pipes are used to separate two different columns. Example can contain many different columns.

#5) Tags:

Cucumber by default runs all scenarios in all the feature files. In real time projects there could be hundreds of feature file which are not required to run at all times.

For instance: Feature files related to smoke test need not run all the time. So if you mention a tag as smokeTest in each feature file which is related to smoke test and run cucumber test with @SmokeTest tag . Cucumber will run only those feature files specific to given tags. Please follow the below example. You can specify multiple tags in one feature file.

Example of use of single tags:

```
@SmokeTest @RegisterTest
Feature: Create Customer Feature
  As a user of XYZ Bank,
  I want to open an account
  So that I have to create an account firstly
  Scenario: Create the customer successfully
    Given I open the website <website>
    Then I verify that add customer screen appears
    When I type first name as <firstName> , last name as <lastName> ,
postcode as <postcode>
    When I click submit button
    Then I check the customer first name as <firstName> , last name as
<lastName> , postcode as <postcode> has been existed in the customer
list

Examples:
| website                    | firstName  | lastName | postcode |
|
http://www.globalsqa.com/angularJs-protractor/BankingProject/#/manager/a
ddCust | Thuy       | Nguyen   | 05550    |
|
http://www.globalsqa.com/angularJs-protractor/BankingProject/#/manager/a
ddCust | Hong     | Ho   | 05147 |
```

Reference http://www.softwaretestinghelp.com/cucumber-bdd-tool-selenium-tutorial-30/