

What is Cucumber

Cucumber is a framework for writing and executing high level descriptions of your software's functionality.

What can you do with Cucumber

Cucumber understands the language Gherkin. It is a Business readable, Domain specific Language that lets you describe software's behavior without detailing how the behavior is implemented. Gherkin serves two purposes – documentation and automated tests.

Gherkin Syntax

Gherkin is a line-oriented language that uses indentation to define structure. Line endings terminate statements (steps). Either space or tabs may be used for indentation. Most lines start with a keyword.

Comments lines are allowed anywhere in the file. They begin with zero or more spaces, followed by a hash sign (#) and some amount of text. The parser divides the input into features, scenario and steps. When you run the feature the trailing portion (after the keyword) of each step is matched to a method (Ruby calls it Step Definitions). An example of a Gherkin source file is shown below:

```
1: Feature: Some descriptive text of what is desired
2:   Textual description of the business value of this feature
3:   Business rules that govern the scope of the feature
4:   Any additional information that will make the feature easier to understand
5:
6:   Scenario: Some determinable business situation
7:     Given some precondition
8:     And some other precondition
9:     When some action by the actor
10:    And some other action
11:    And yet another action
12:    Then some testable outcome is achieved
13:    And something else we can check happens too
14:
15:   Scenario: A different situation
16:     ...
```

First line starts the feature.

Lines 2-4 are not parsed by the parser. (These lines are expected to describe the business value of this feature.

Line 6 starts a scenario.

Lines 7-13 are the steps for the scenario.

Line 15 starts the next scenario and so on.

Document reference for Gherkin: <https://cucumber.io/docs/reference>

Link for downloading Eclipse plugin: <http://cucumber.github.com/cucumber-eclipse/update-site>

How to Install Cucumber

Following steps are needed to install Cucumber in eclipse. A sample install will be demonstrated in the class.

1. Create a new project in eclipse {For example: Cucumber_project}
2. Add jar files
3. Add feature file
4. Run feature file
5. Create testRunner Class for running feature file
6. Add Selenium script

Create a new project in Eclipse

From the Package Explorer: File -> New -> Java Project [Name it: **Cucumber_Project**]

Right Click on the newly created project -> Build Path -> Configure Build Path...

Click "Libraries" on the top then Click on "Add External JARs..." on the right

It will open up the Windows Explorer. Navigate to the folder where you have kept your Cucumber jar files.

Select all the jar files and Click "Open". This will add all the jar files to the Library.

Click "Apply and Close".

You are done with this part.

Adding jar files

cobertura-2.1.1.jar

cucumber-core-2.3.1.jar

cucumber-java-2.3.1.jar

cucumber-junit-2.3.1.jar

cucumber-jvm-deps-1.0.6.jar

cucumber-reporting-0.0.21.jar

gherkin-4.1.3.jar

junit-4.12.jar

mockito-all-2.0.2-beta.jar

selenium-server-standalone-3.5.3.jar

Adding feature file

Right Click on "Cucumber_project" -> New -> Folder [A new folder will be created: name it "**features**"]

Right Click on the "features"-> New -> File [A new file will be created: name it "**SampleApplication.feature**"]

At this point a feature file [named: SampleApplication.feature] has been added to the project.

Right Click on the feature file -> Open With -> Cucumber Editor [This will open up an editor on the right side]

Copy the following text to the editor:

Feature: Test Facebook smoke scenario

Scenario: Test login with valid credentials

Given Open firefox and start application

When I enter valid username and valid password

Then user should be able to login successfully

Save it [CTRL s]