5.5 Gherkin Comments and Tags

****

This section will guide you to understand:

* What is Gherkin Comment
* What are Gherkin Tags

**Development Environment:**

* JRE: OpenJDK Runtime Environment 11.0.2
* Eclipse IDE for Enterprise Java Developers v2019-03 (4.11.0)
* TestNG
* Selenium jars
* Cucumber jars

This guide has three subsections, namely:

5.5.1 Gherkin comments with example

5.5.2 Gherkin tags with example

5.5.3 Pushing the code to your GitHub repositories

**Step 5.5.1:** Gherkin comments with example.

* Comment is basically a piece of code meant for documentation purposes and not for execution.
* Feature file: In case of a feature file, we just need to put # before beginning your comment.

**Example:**

|  |
| --- |
| Feature: annotation  #This is how background can be used to eliminate duplicate steps  Background:  User navigates to Facebook  Given I am on Facebook login page |

* Step definition file: If you are using Java as a platform then mark your comments with //.

**Example:**

|  |
| --- |
| //scroll to the bottom of the page  ((JavascriptExecutor) driver).executeScript("window.scrollTo(0, document.body.scrollHeight)"); |

**Step 5.5.2:** Gherkin tags with example.

* if we have many scenarios in a feature file, to put them under a single umbrella, we use tags to generate reports for specific scenarios under the same tag.
* Tags are defined in our runner class like this:

|  |
| --- |
| @RunWith(Cucumber.class)  @CucumberOptions{  format= {"Pretty" ,"json:target/output.json", "html:targer/html"},  feature={"src/functional-test/resources"},  tags={"@tag", "@tag1"} } |

* When we define multiple tags in runner class in below form, it will be defined with the use of logical operator:
  + tags = {“@tag”, “@tag1”}: means AND condition. It says that scenarios matching both these tags need to be executed.
  + tags = {“@tag1, @tag2”}: means OR condition. It says that scenarios matching any of this tag need to be executed.

**Step 5.5.3:** Pushing the code to GitHub repositories.

Open your command prompt and navigate to the folder where you have created your files

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add . 

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master