5.6 Step Definition****

This section will guide you to understand:

* What is cucumber step definition
* Implementation of Cucumber in Selenium

**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:

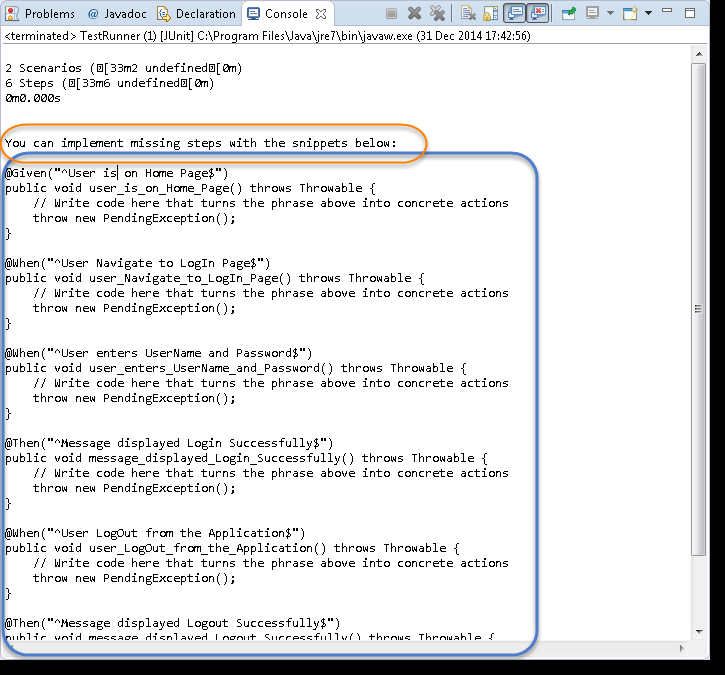
5.6.1 Adding a step definition file

5.6.2 Providing implementations using Selenium Java for the step definition methods

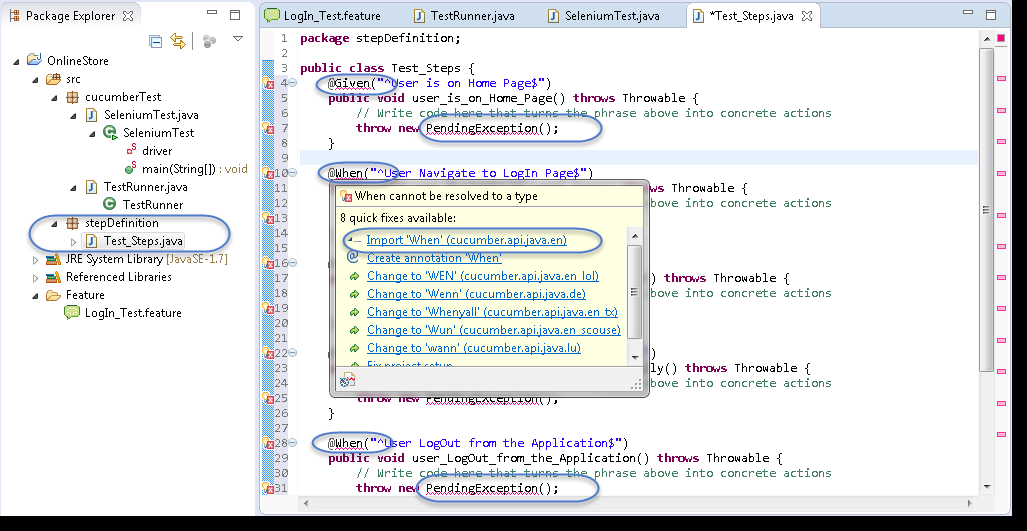
5.6.3 Pushing the code to your GitHub repositories

**Step 5.6.1** Adding a step definition file.

* Create a new Class file in the “stepDefinition” package and name it as “Test\_Steps”
* When we run the Feature file as Cucumber feature, the following message will be displayed in the console window:



* Copy the complete text marked in a blue box and paste it into the above created Test\_Steps class.
* The test will show many errors on “@” annotations. At the annotations, import the “cucumber.api.java.en” for all the annotations.



## 

## **Step 5.6.2:** Providing implementations using Selenium Java for the Step Definition methods.

* Using Selenium Java, provide the implementation to the first method “@Given (“^User is on Home Page$”)”
* Launch the Browser
* Navigate to Home Page

Method will look like this now:

|  |
| --- |
| **@Given("^User is on Home Page$")**  **public void user\_is\_on\_Home\_Page() throws Throwable**  **{**  **driver = new FirefoxDriver();**  **driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS); driver.get("http://www.store.demoqa.com");**  **}** |

## 

* Using Selenium Java, provide the implementation to the second method “@When (“^User Navigate to Login Page$”)”
* Click on the Login link

Method will look like this now:

|  |
| --- |
| @When("^User Navigate to Login Page$")  public void user\_Navigate\_to\_Login\_Page() throws Throwable  {  driver.findElement(By.xpath(".//\*[@id='account']/a")).click();  } |

* Using Selenium Java, provide the implementation to the second method.

“@When (“^User enters UserName and Password$”)”

* Enter UserName and Password
* Click on Submit button

Method will look like this now:

|  |
| --- |
| @When("^User enters Username and Password$")  public **void** user\_enters\_Username\_and\_Password() throws Throwable  {  driver.findElement(By.id("log")).sendKeys("testuser\_1");  driver.findElement(By.id("pwd")).sendKeys("Test@123");  driver.findElement(By.id("login")).click();  } |

* Do the same steps for the rest of the methods and the complete Test\_Steps class will look like this:

|  |
| --- |
| **package stepDefinition;**    **import java.util.concurrent.TimeUnit;**  **import org.openqa.selenium.By;**  **import org.openqa.selenium.WebDriver;**  **import org.openqa.selenium.firefox.FirefoxDriver;**  **import cucumber.api.java.en.Given;**  **import cucumber.api.java.en.Then;**  **import cucumber.api.java.en.When;**  **public class Test\_Steps**  **{ public static WebDriver driver;**  **@Given("^User is on Home Page$")**  **public void user\_is\_on\_Home\_Page() throws Throwable{**  **driver = new FirefoxDriver();**  **driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);**  **driver.get("http://www.store.demoqa.com"); }**    **@When("^User Navigate to LogIn Page$")**  **public void user\_Navigate\_to\_LogIn\_Page() throws Throwable {**  **driver.findElement(By.xpath(".//\*[@id='account']/a")).click(); }**    **@When("^User enters UserName and Password$")**  **public void user\_enters\_UserName\_and\_Password() throws Throwable {**  **driver.findElement(By.id("log")).sendKeys("testuser\_1");**  **driver.findElement(By.id("pwd")).sendKeys("Test@123");**  **driver.findElement(By.id("login")).click(); }**    **@Then("^Message displayed Login Successfully$")**  **public void message\_displayed\_Login\_Successfully() throws Throwable{**  **System.out.println("Login Successfully"); }**    **@When("^User LogOut from the Application$")**  **public void user\_LogOut\_from\_the\_Application() throws Throwable{**  **driver.findElement (By.xpath(".//\*[@id='account\_logout']/a")).click(); }**    **@Then("^Message displayed Logout Successfully$")**  **public void message\_displayed\_Logout\_Successfully() throws Throwable {**  **System.out.println("LogOut Successfully"); }}** |

**Step 5.6.3:** Pushing code to your 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