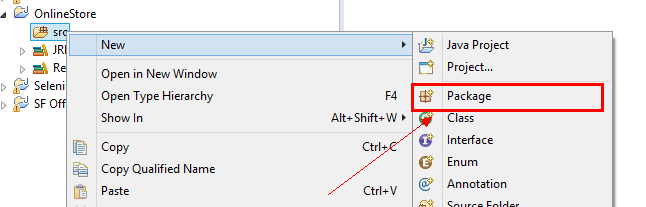
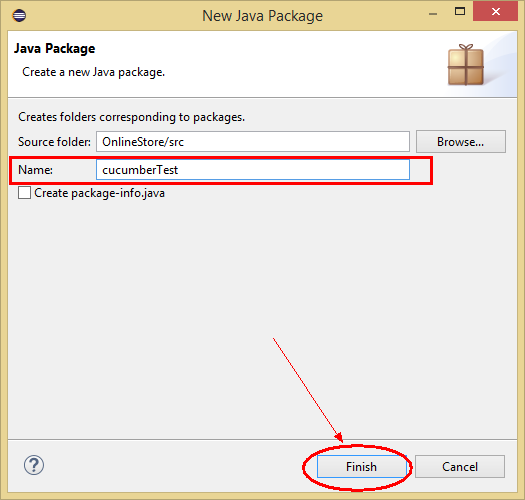
***Getting Started with Cucumber Selenium Java Test***

***Create Folder Structure***

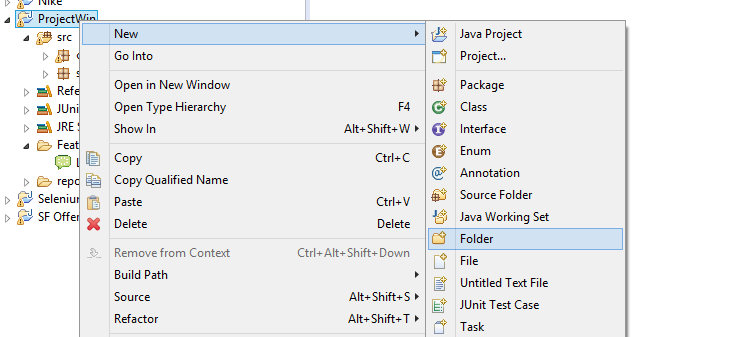
1. *Create a new* ***Package*** *by right click on the ‘****src****‘ folder and select* ***New > Package.***



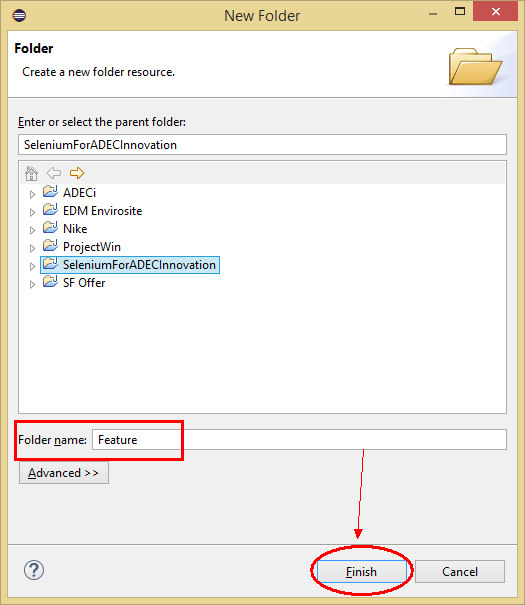
1. *Name it as ‘****cucumberTest****’ and click on* ***Finish*** *button*



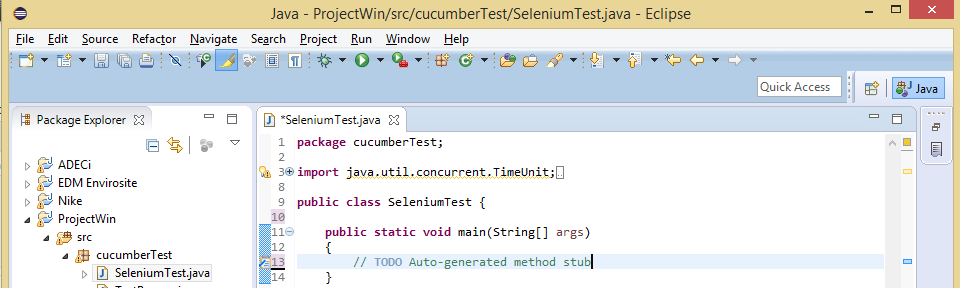
1. *Create another* ***Package*** *and name it as ‘****stepDefinition’****, by right click on the ‘****src****‘ folder and select* ***New*** *>* ***Package****.*
2. *Create a new* ***Folder*** *this time by right click on the project ‘****ProjectWin****‘ and select* ***New*** *>* ***Folder****.*



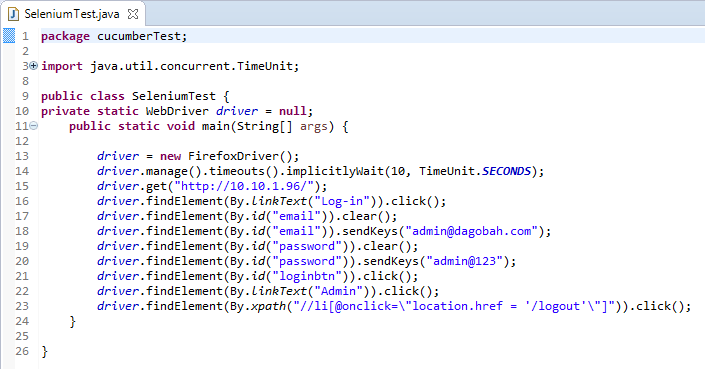
1. *Name it as ‘****Feature****’ and click on* ***Finish*** *button*



1. *Create a new* ***Class*** *file in the ‘****cucumberTest****‘ package and name it as ‘****SeleniumTest****‘, by right click on the Package and select* ***New*** *>* ***Class****. Check the option ‘****public static void main****‘ and click on* ***Finish*** *button.*



***Selenium Test Script : Log In and Log Out***

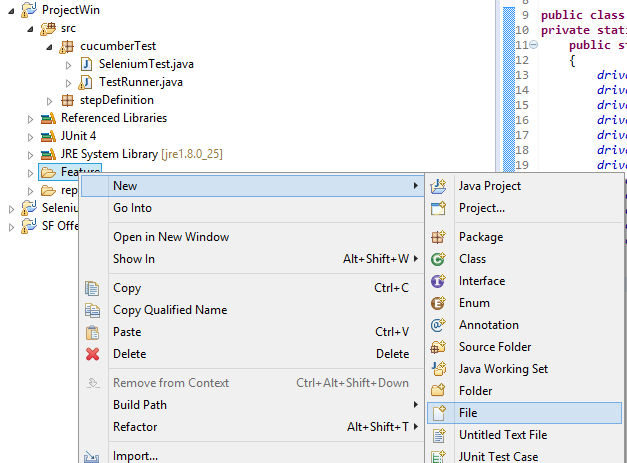


*Now, to start the test just select* ***Run*** *>* ***Run******As*** *>* ***Java******Application*** *Or Right Click on Eclipse code and Click* ***Run******As*** *>* ***Java******Application****. After a few Seconds a* ***Mozilla*** *browser will open and you will see that with the help of your script, Selenium will Launch the Online Store demo application, perform* ***Sign in****.*

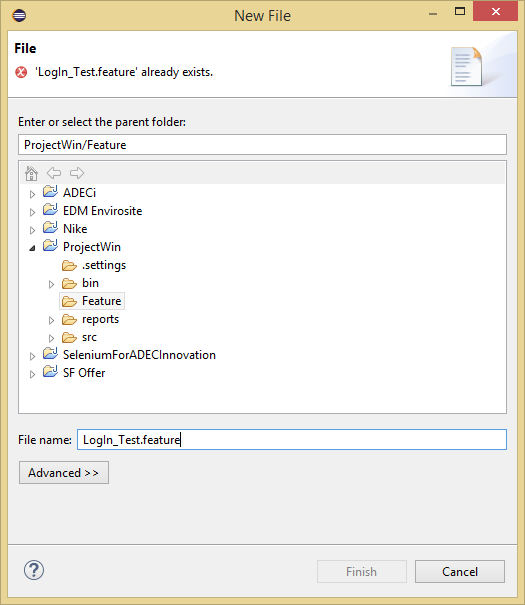
***Cucumber Test Script (Feature File)***

*A* ***feature*** *file is an entry point to the Cucumber tests. This is a file where you will describe your tests in Descriptive language (Like English). It is an essential part of* ***Cucumber****, as it serves as an automation test script as well as live documents. A* ***feature*** *file can contain a scenario or can contain many scenarios in a single feature file but it usually contains a list of scenarios.*

1. *On the* ***Feature*** *folder Right click and select* ***New*** *>* ***File***

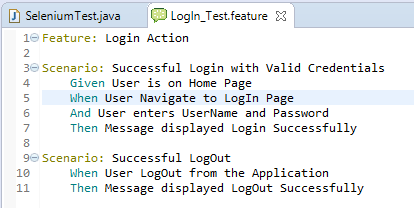


1. *In order for* ***Cucumber*** *to automatically detect the stories (or* ***features****, as they’re known in Cucumber), you need to make sure that they carry the ‘.****feature****‘ file extension.*



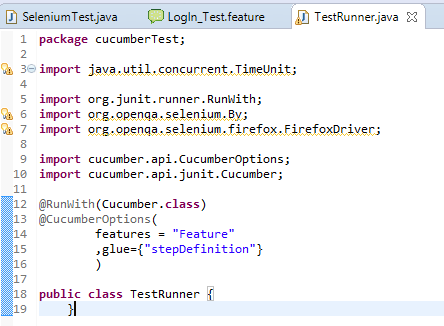
1. *Write your first cucumber script now*

***Cucumber Test Script***



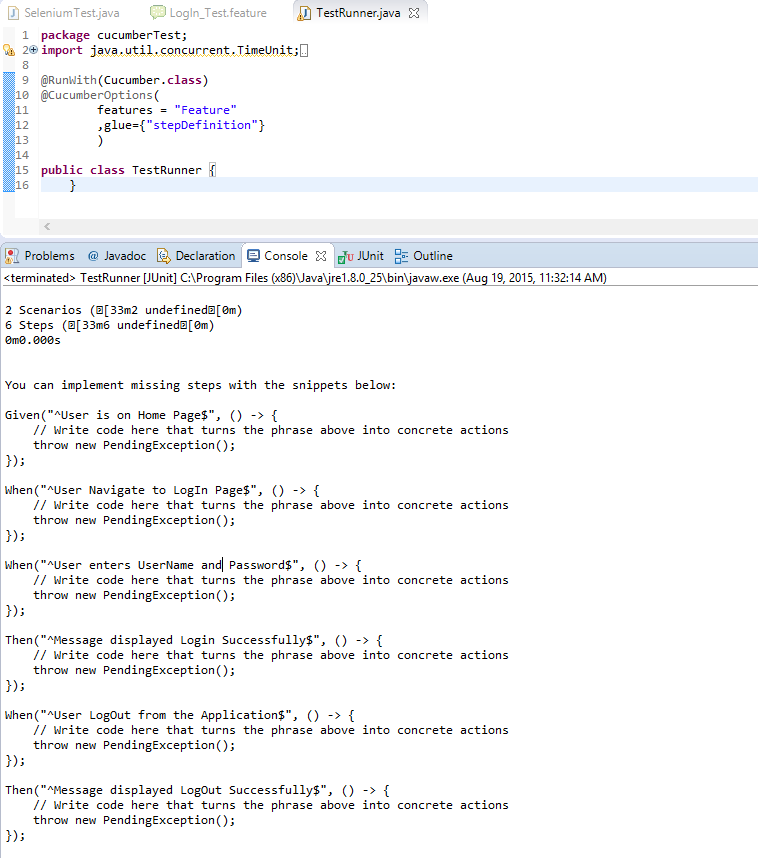
***Junit Test Runner Class***

*As* ***Cucumber*** *uses* ***Junit*** *we need to have a* ***Test Runner class****. This class will use the Junit annotation* ***@RunWith(),*** *which tells JUnit what is the test runner class. It more like a starting point for Junit to start executing your tests. In the src folder create a class called* ***TestRunner****.*



***Run the Cucumber Test***

*Click on the Run button on eclipse and you have your test run Or Right Click on* ***TestRunner*** *class and Click* ***Run******As*** *>* ***JUnit******Test******Application****.*

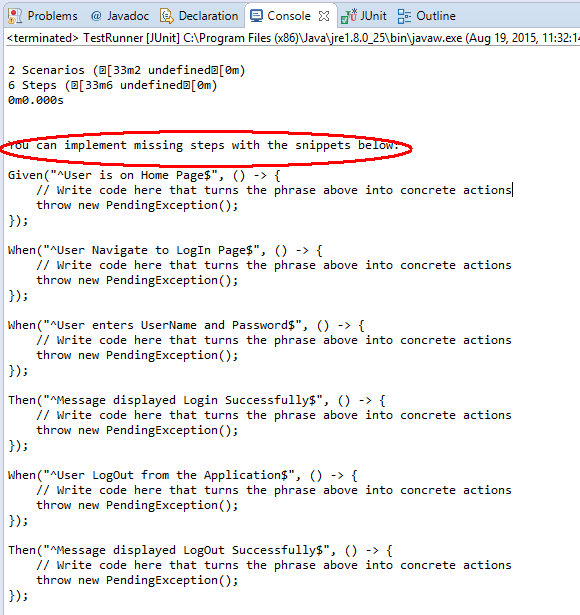


***What is Step Definition?***

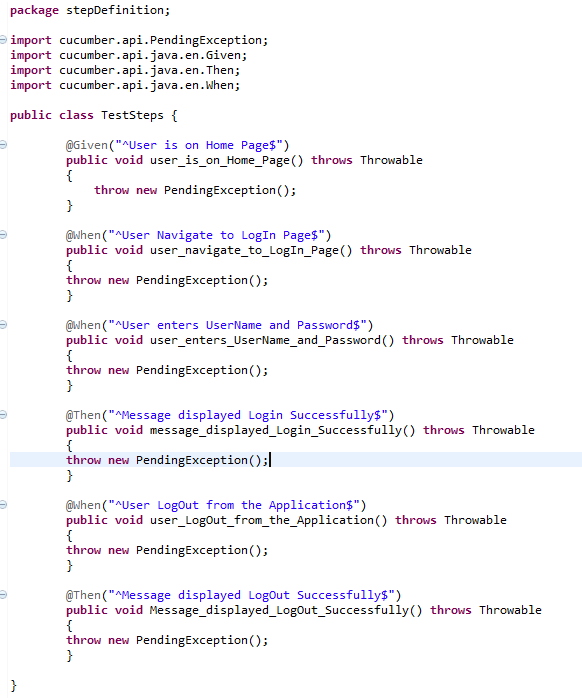
***Step Definition*** *is a small piece of code with a pattern attached to it or in other words a Step Definition is a java method in a class with an annotation above it. An* ***annotation*** *followed by the pattern is used to link the Step Definition to all the matching Steps, and the code is what Cucumber will execute when it sees a* ***Gherkin******Step****. Cucumber finds the Step Definition file with the help of* ***Glue******code*** *in* ***Cucumber******Options****.*

***ADD a Step Definition File***

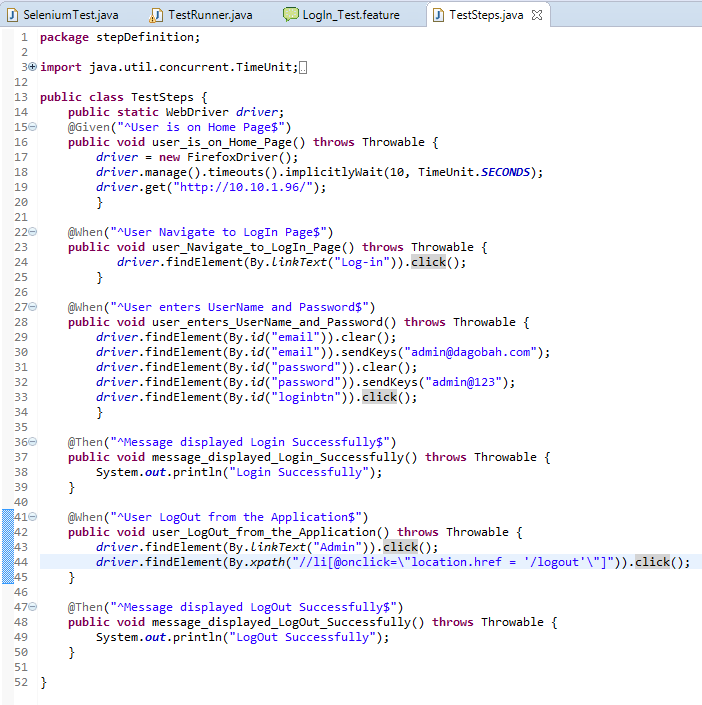
1. *Create a new* ***Class*** *file in the ‘****stepDefinition****‘ package and name it as ‘****Test****\_****Steps****‘, by right click on the* ***Package*** *and select* ***New*** *>* ***Class****. Do not check the option for ‘****public******static******void******main****‘ and click on* ***Finish*** *button.*
2. *Take a look at the message in the console window. This message was displayed, when we ran the* ***TestRunner*** *class.*



1. *Notice, the eclipse console window says ‘****You can implement missing steps with the snippets below:****‘. It is very easy to implement all the steps, all you need to do is to copy the complete text marked in a blue box and paste it in to the above created* ***TestSteps*** *class. As of now the test will show many errors on* ***‘@‘******annotations****. Mouse hover at the annotations and import the ‘****cucumber.api.java.en****‘ for all the annotations.*

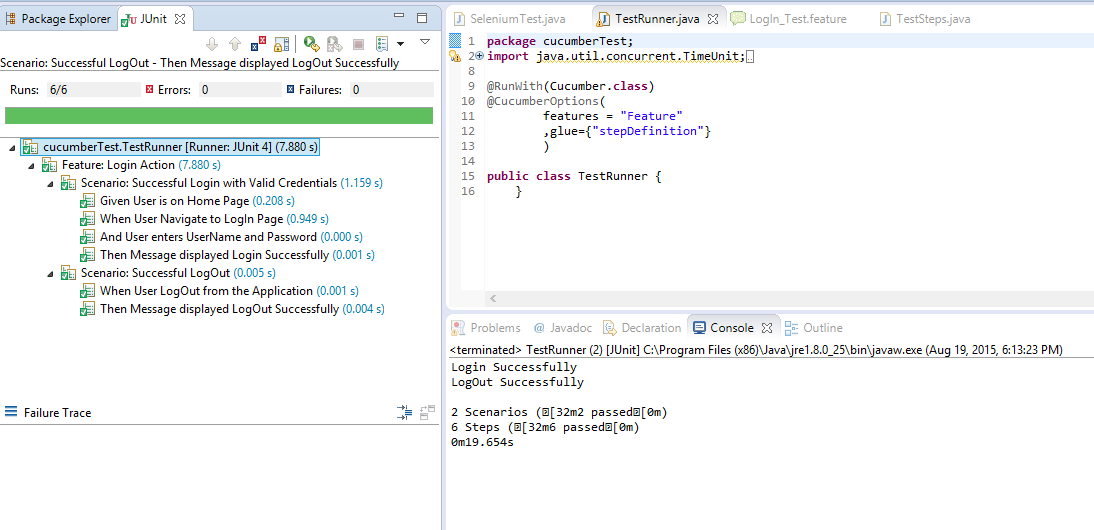


***ADD Selenium Java Code in the Step Definition Methods***



***RUN the Cucumber Test***

*Right Click on* ***TestRunner*** *class and* ***Click******Run******As*** *> J****U****nit* ***Test****. Cucumber will run the script the same way it runs in* ***Selenium******WebDriver*** *and the result will be shown in the left hand side* ***project******explorer******window*** *in* ***JUnit*** *tab.*

**

***Cucumber*** *starts it’s execution by reading the* ***feature******file*** *steps. As soon as* ***Cucumber*** *reaches to the first* ***step*** *for e.g.* ***Given*** *statement of* ***Scenario****, it looks for the same statement in the* ***Step******Definition*** *file, the moment it find the statement, it executes the piece of code written inside the function.*