Cucumber

Cucumber is a frame work which is used for implementing BDD(Behavior - driven development).

In BDD automation programs are created based on the behavior of the Application and not based on the Test data.

Cucumber was initially implemented with Ruby later it was extended to work on Java and C#.

In Cucumber the automation programs are created based on a file called feature file.

In this feature file the task to be automated is written in plain English. These English statements are connected to the Selenium Program which perform those activities.

The advantage of this process is the flow of the Automation Programs can be understood by Non-technical person.

Note: - Cucumber does not support TestNG, it works on Junit frame work.

Cucumber uses its own language called as Gherkin. This works on certain annotations.

Feature:- This represents the module or functionality that is under Test

Scenario:- This represents the test case that is been automated in a particular Feature.

One feature can have multiple Scenario's.

Generally the title of the Test case is given as Scenario.

Given:- This represents the Pre-Condition of the Test case.

When:- This represents the exact action that is performed in the Test Case.

And:- This represents any additional actions that should be performed on the Test case.

Then:- This represents the out come of the Test Case.

But:- This represents any negative condition that should be tested.

Important Annotations in Junit

@Test: - This represents a Test case

@Before:- This is executed prior to each Test case on the current class. It is similar to @Before Method in TestNG.

@After:- This is executed after each Test case in the current class. This is similar to @After Method

@Before Class:- This is executed before each class in the current Test Suite

@After Class:- This is executed after each class in the current Test Suite

@RunWith:- This is used to specify what kind of Program under Execution.

Eg: Is it a Test suite of cucumber program

@SuiteClasses:- This is used by Junit for performing Test Suite execution. The name of all the classes that should be executed as a Test Suite should be passed as Arugments to this Annotation.

Configuring Junit into Eclipse Project

Right Click on the Project in Eclipse --> Click on Properties --> Click on Java Build path --> Libraries --> Add library --> Click on Junit --> Click on Next --> Select Junit4 from drop down --> Click on Finish.

:BDD:

1. Add jars
2. Add feature file
3. Run feature file
4. Create testRunner for running feature file
5. Add Selenium script (with cucumber)

Goto Eclipse --> create a java Project --> Add all the jar files to the project--> Add selenium server jar files too

Right click on the Project --> New --> Folder --> give name as **features(** inside this we will write all the feature files)

Goto Help menu --> Install new Software (search for this pulug-in and install -- http://cucumber.github.com/cucumber-eclipse/update.site) .. it will ask for eclipse to restart

Right Click on the "features folder" --> file --> give the file name (ApplicationTest.feature) --> click on finish. Now write the feature file conditional statements

```
Feature: Test newTours Application
Scenario: Test login with valid credentials
Given open firefox and start Application
When I enter valid userName and Password
Then user should be able to login successfully
```

Now right click on the feature file created --> Run as Cucumber feature (when we run this we wont get anything because its just a plain English text)

To execute a feature file we need a TestRunner.

Now create a package --> create a class(TestRunner) , no public static void main()

In this TestRunner we specify goto "features" and run the "feature file created"