***Cucumber:***

*Cucumber is a testing framework which supports Behaviour Driven Development (BDD). It lets us define application behaviour in plain meaningful English text using a simple grammar defined by a language called Gherkin. Cucumber itself is written in Ruby, but it can be used to “test” code written in Ruby or other languages including but not limited to Java, C# and Python.*

**Behaviour Driven Development**

* *Tests are written in plain descriptive English type grammar*
* *Tests are explained as behaviour of application and are more user-focused*
* *Using examples to clarify requirements*

### *Example of a Cucumber/SpecFlow/BDD Test:*

The main feature of the Cucumber is that it focuses on Acceptance testing. It made it easy for anyone in the team to read and write test and with this feature it brings business users into the test process, helping teams to explore and understand requirements.

***Feature: Sign up***

***Sign up should be quick and friendly.***

***Scenario: Successful sign-up***

***New users should get a confirmation email and be greeted personally by the site once signed in.***

***Given I have chosen to sign up***

***When I sign up with valid details***

***Then I should receive a confirmation email***

***And I should see a personalized greeting message***

***Scenario: Duplicate email***

***Where someone tries to create an account for an email address that already exists.***

***Given I have chosen to sign up***

***But I enter an email address that has already registered***

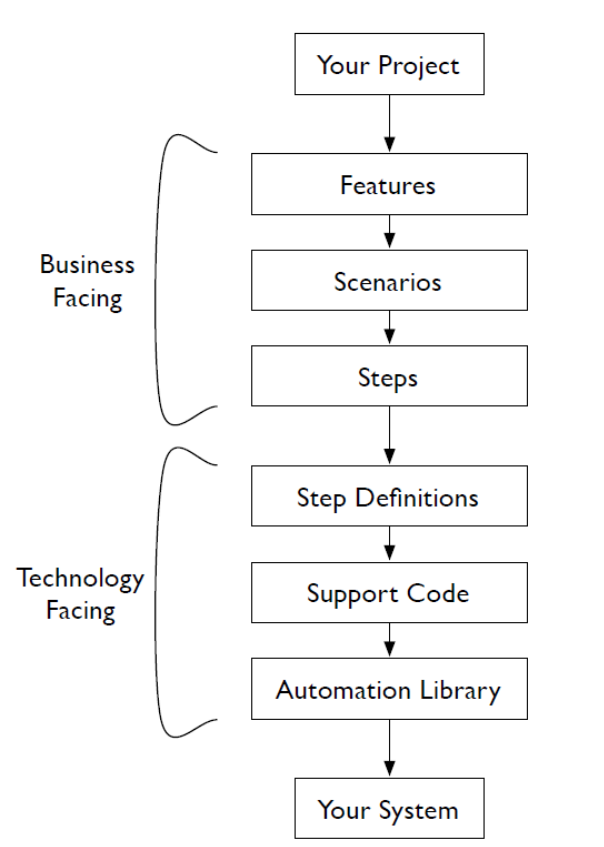
***Then I should be told that the email is already registered***

***And I should be offered the option to recover my password***

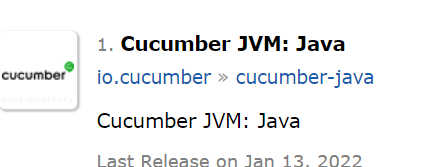
Now take a look at the above example code anybody can understand the working of the test and what it is intended to do. It gives an unexpected powerful impact by enabling people to visualize the system before it has been built. Any of the business users would read and understand the test and able to give you feedback that whether it reflects their understanding of what the system should do, and it can even lead to thinking of other scenarios that need to be considered too.

When working with BDD there are a couple of additional concepts we need to be familiar with:

* **FeatureFile** – the same Spec document that describes the system’s behavior. In this file, we will write the Gherkin language with the Feature, the Scenarios, and with the keywords mentioned above.
* **Step Definitions** – a class or collection of classes in which we translate the Gherkin language into code that executes automatic actions on the tested product.



1. Navigate to <https://mvnrepository.com/>
2. And search for cucumber java and click on search
3. Click on below



1. Select version as show in below



1. Copy below add it in POM.xml file as dependency

<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->

<dependency>

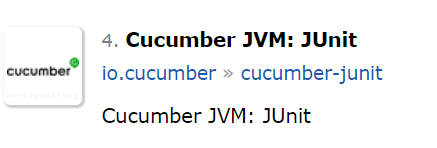
<groupId>io.cucumber</groupId>

<artifactId>cucumber-java</artifactId>

<version>6.9.1</version>

</dependency>

1. Repeat step 1 and search for cucumber junit and select as per below



1. **Select same version that is selected for step 4**



1. **Copy below and paste in POM.xml file as shown below**

<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-junit -->

<dependency>

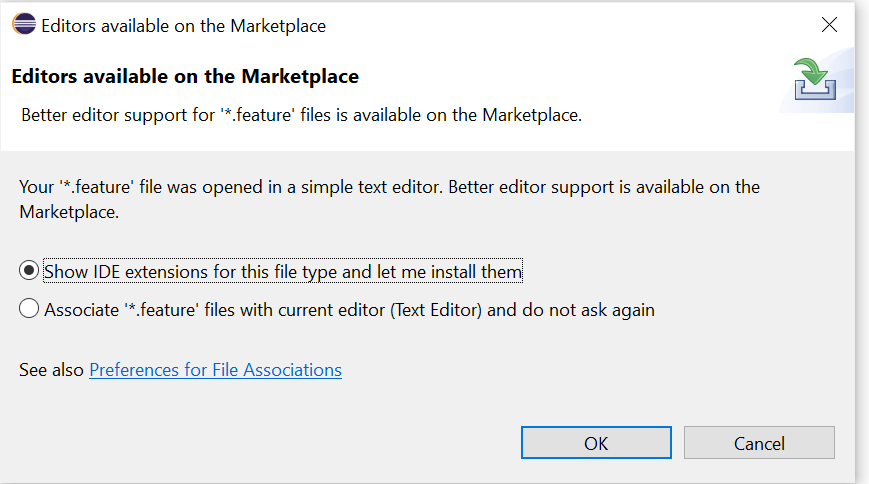
<groupId>io.cucumber</groupId>

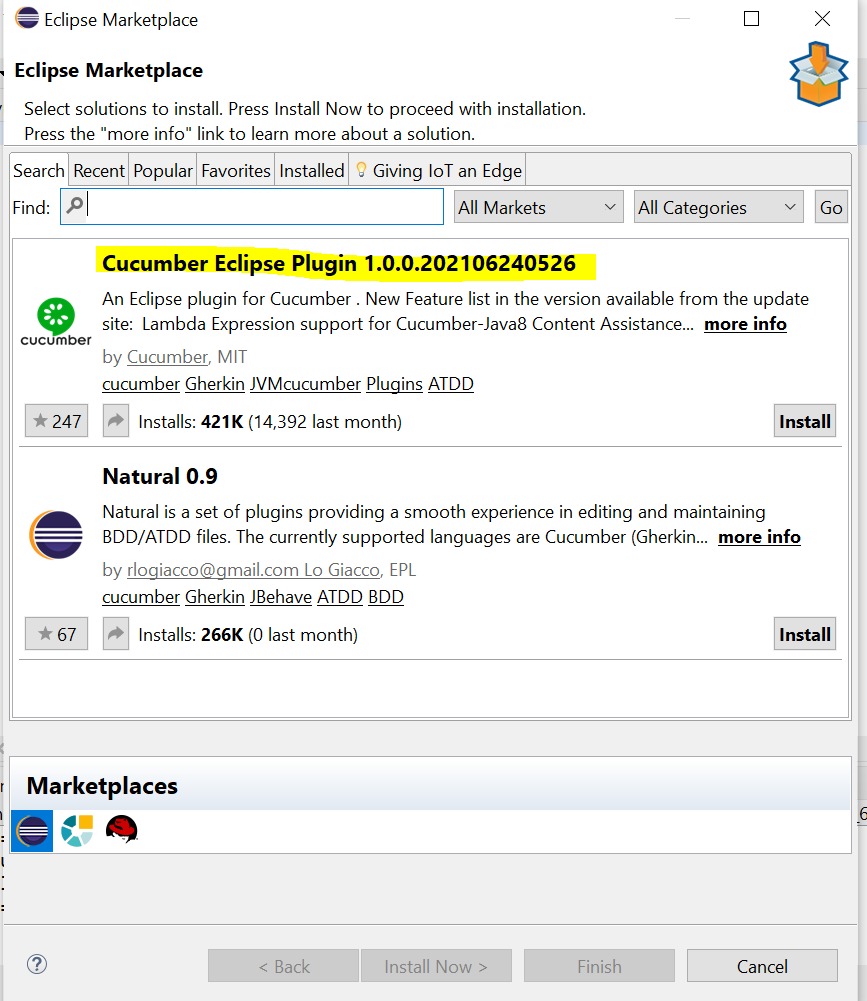
<artifactId>cucumber-junit</artifactId>

<version>6.9.1</version>

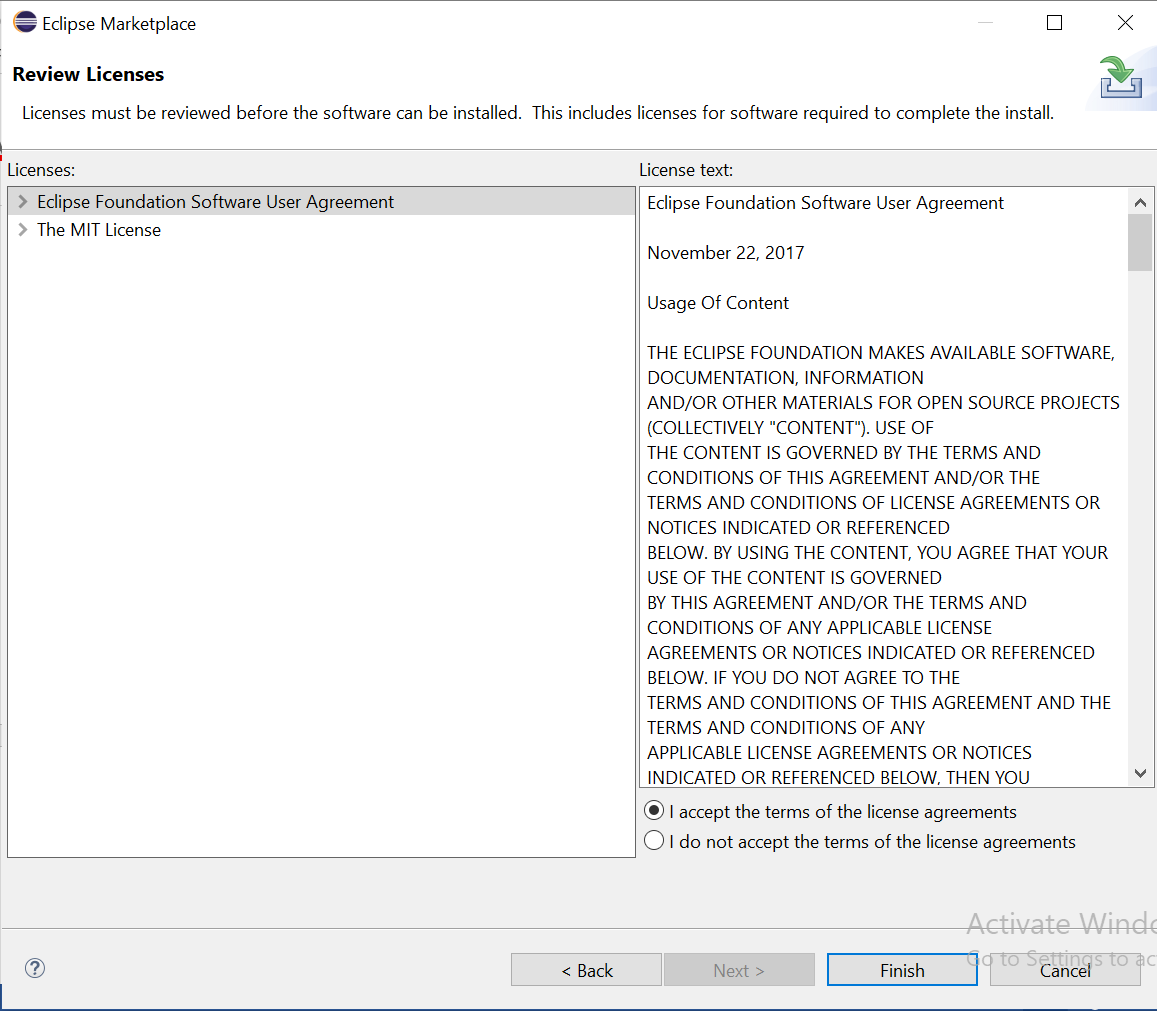
<scope>test</scope>

</dependency>

1. Right click on your project in Eclipse and create new folder with name as Features
2. Right click on Features folder and create new file with name sprint1.feature
3. Click Ok on below screen
4. Click on Install from below screen



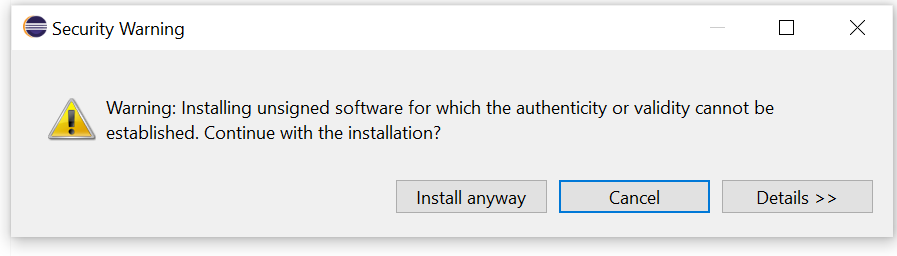
1. Click on I accept



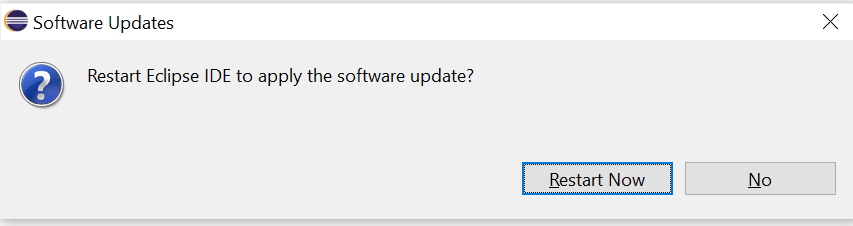
1. Wait until installation is completed



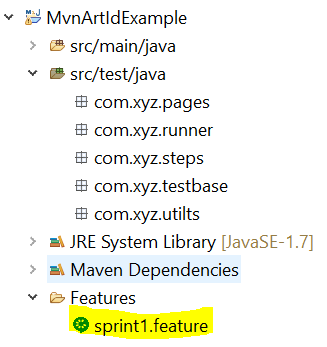
1. Click on Install anyway



1. Click on Restart Now



1. Open sprint.feature file



1. Paste below code in sprint.feature file

Feature: HRM Login Test Case

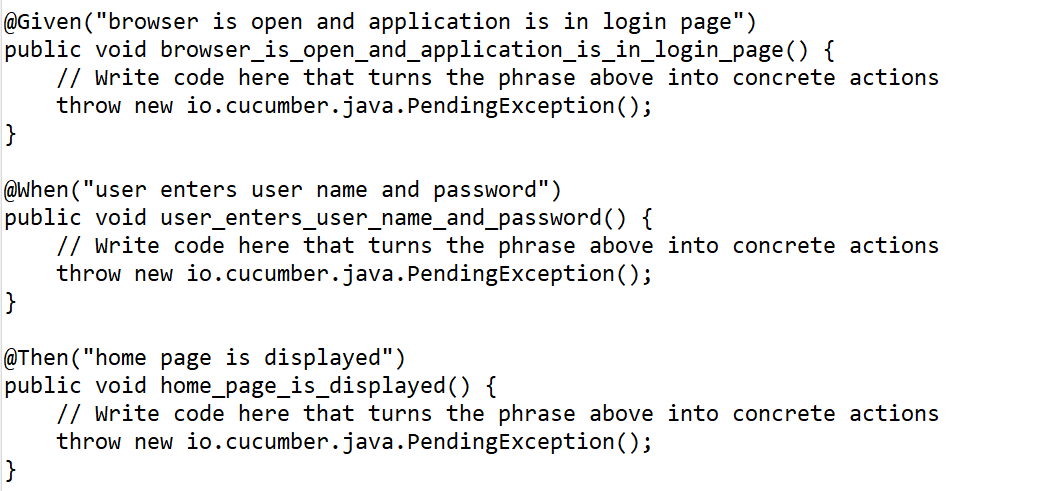
Scenario: Verify user is able to login with valid user id and password

Given browser is open and application is in login page

When user enters user name and password

Then home page is displayed

1. Run this future file by selecting runas->cucmber feature
2. Copy below code



1. Create a new package called “stepdefinations” under src/test/java in eclipse
2. Create new class in above package
3. Paste code which is copied in step 20
4. Than add code for all the methods from step 20
5. Open future file and run this file to execute your code