* **Cucumber Framework**:{ Behavior driven development; BDD}. The whole testing is done based on User Story.
* **To Download:**

Goto eclipse Marketplace

In search box, type – Cucumber & click install

Accept any Terms

* **Add dependencies:**

Goto Maven Repository

Search box – Cucumber.

Needs 3 type of dependencies

*1.Cucumber JVM:Java;V7.15x*

*2.Cucumber JVM:Junit 4; V7.15*

*3.Cucumber JVM:Core; V7.15*

*4.Note You also need a Driver; V4.16.1*

*5. Screenshot (Apache Commons IO); V2.15.1*

Copy the dependency & paste in POM file

* **Cucumber; Folder Organizing:**

1. loginFeature.feature **----🡪** src/test/resources (Need to create a *folder, then file*)
2. loginStepDefination **-----🡪** src/test/java (Need to create a *Package, then class*)
3. testRunner **-------🡪** src/test/java (Need to create a *Package, then class*)
4. PageObejectModels **-----🡪** src/test/java

* (inside package) - BaseClass, loginPage, ProductPage, CartPage, CheckOutPage, LogoutPage

A screenshot of a computer program

Description automatically generated

***In detail:***

1. **loginFeature.feature (***src/test/resources***):** Used to store User stories. Create a file **[***FeatureFiles***]**. Inside create a file ---->**[***filename.feature***]**

* **Gherkin Law**
  + **Author**: this is You
  + **Feature**: List of scenarios
  + **Scenarios**: List of steps with Arguments
  + **Given**: Precondition step {browser opens and need to be on the page}
  + **When**: key Action {enter Username}
  + **And**: key Action {Used after when, more used for action; enter Password}
  + **Then**: the solution {I should land product page}
  + **Scenario Outline**: Used to contain data’s
  + **Background**: steps will run before each scenario (like TestNG: Annotation)
* Example: Scenario

Feature: Sauce Demo Login Features

Scenario: Login with valid username and password

Given User is on the Saucedemo.com login page

And User enter correct username

And User enter correct password

And User click Login button

Then User will be taken to the product page

***Right click 🡪 Run As 🡪 Cucumber Feature***

Console should print out Methods of 5 action created in feature file;

*Note: And will not convert in method however it can be adjusted*

* Example of Scenario Outline:

Scenario Outline: Login with different Username and Password

Given user is on the Saucedemo.com login page

When User enter "<Username>" into username field

And User enter "<Password>" into the password field

And Click Login button

Then user will be taken to the product page

Examples:

| Username | Password |

| standard\_user | secret\_sauce |

| problem\_user | secret\_sauce |

1. **loginStepDefination (***src/test/java***):** Inside this folder all the methods printed form feature files should store here. That is where you perform actions from POM class
2. **testRunner (***src/test/java***):** This is where we glue feature file & step definition class together and execute.
   * + We will not write anything inside the class (advance level)
     + ON TOP OF CLASS
       - @RunWith(Cucumber.**class**)
       - @CucumberOptions**(***Everything goes inside this Annotations /hooks*

* features ="src/test/resources/FeatureFiles/loginTestFeature.feature",
* glue = "StepDefinitions"
* plugin = {"pretty", "html:target/Cucumberreport.html"}
* monochrome = **false**, goes hand to hand with {dryRun =}
* dryRun = **true**

***)***

1. **PageObjectModel** **(***src/test/java***)**: where the outline for all classes. Will be used to create methods inside StepDefination classes to execute the user stories.

* **How to group TC’s in Cucumber using Tags & run as a suite** 
  + In cucumber you use tags
    - * @smoke , @ regression : on feature.file (how to group)
      * In a TestRunner class tags🡨