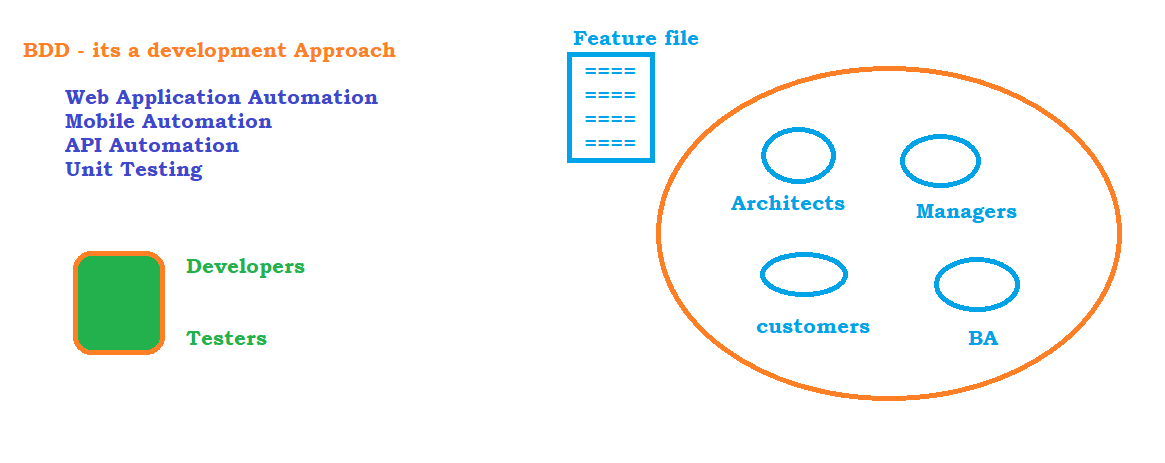
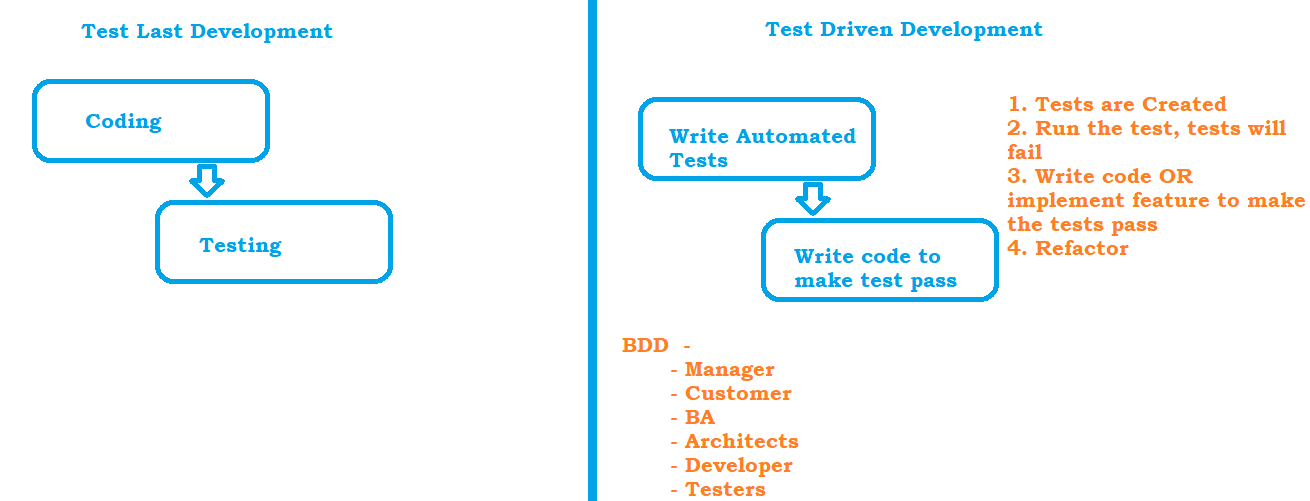
BDD Cucumber

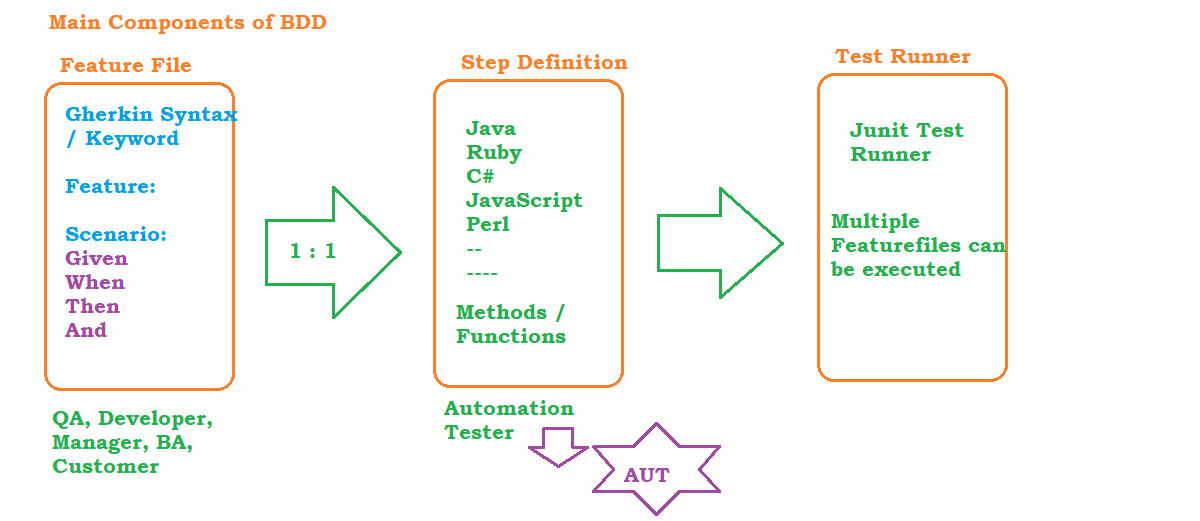
Official Web site - <https://cucumber.io/>



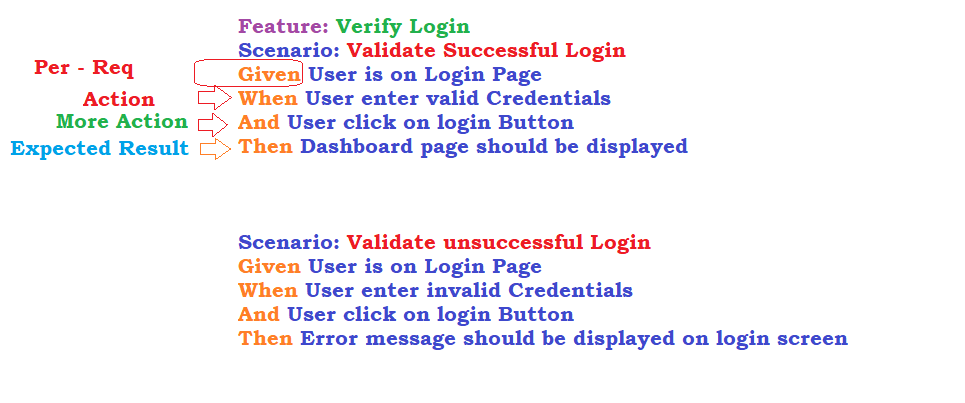
# Development Approach :



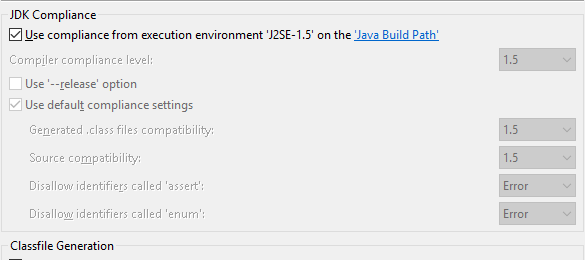
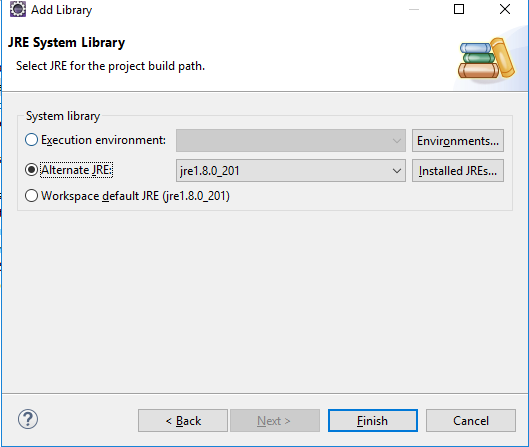
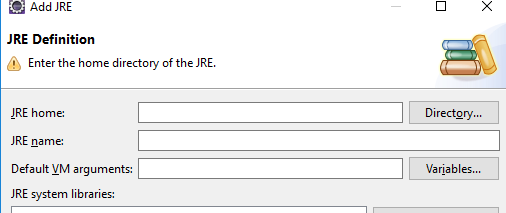
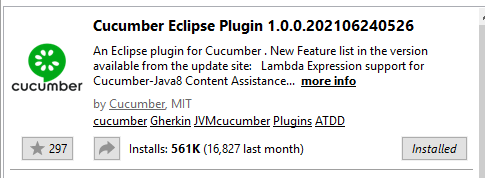
BDD – Components



# Sample Feature File :



Project Setup in Eclipse

* Download and install JDK1.8
  + <https://www.oracle.com/in/java/technologies/javase/javase8-archive-downloads.html>
* Update Environment Variables
  + JAVA\_HOME - C:\Program Files (x86)\Java
  + PATH - C:\Program Files (x86)\Java\jdk1.8.0\_341\bin
* Download and install Eclipse
  + <https://www.eclipse.org/downloads/>
  + <https://www.eclipse.org/downloads/packages/release/2022-09/r/eclipse-ide-java-developers>
* Create Maven Project
  + File-> New Maven Project
  + 
* Update JDK to what is installed in the system
  + Right click -> go to properties -> Java compiler -> update the JDK to 1.8 anc click on apply and close
  + 
* Update JRE to what is installed in the system and also JRE should be from JDK Folder
  + Right click -> go to properties -> Java build path ->
  + Click on libraries
  + Select JRE-1.5 and click on Remove button
  + Click on Add Library
  + Click on JRE System library
  + 
  + Click on installed JRE -> Click on ADD button
  + Standard VM
  + Browse and select JRE From JDK location
    - 
  + Apply and close
* Update POM.xml with the required dependencies
  + cucumber-java
  + cucumber-junit
  + junit
  + selenium
* Cucumber Plugin
  + Help -> Eclipse Market Place
  + 

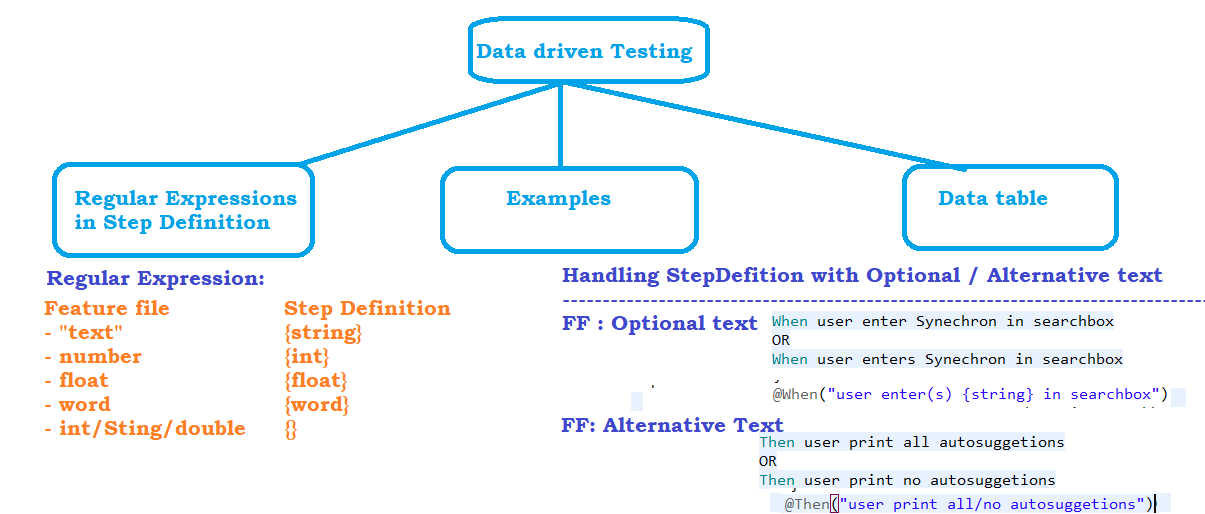
## Automate Driver Executables:

* update pom.xml with webdrivermanager library
* <https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager>

Data driven testing in BDD

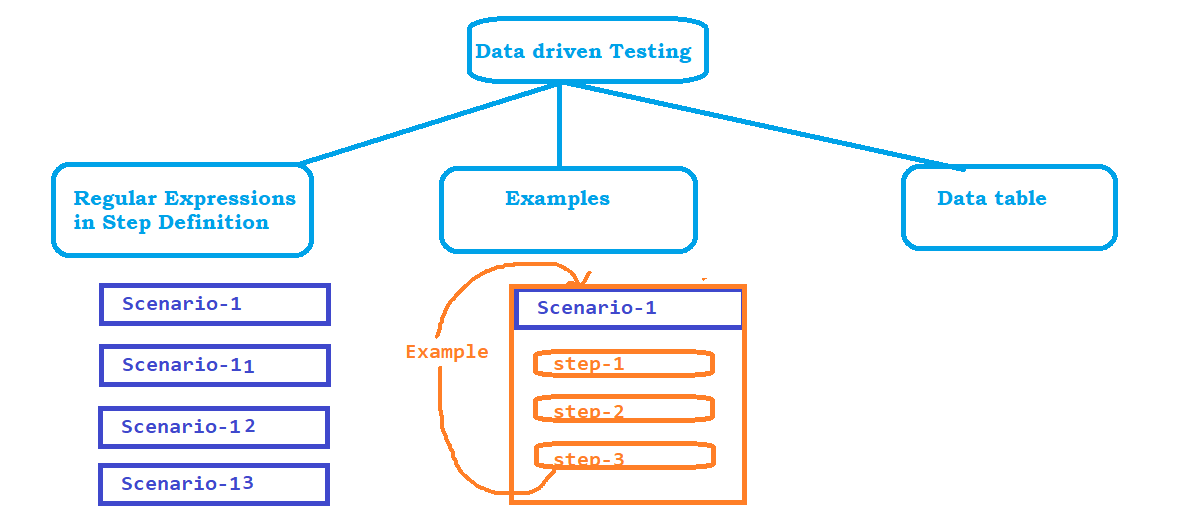
# Regular Expressions:

-------------------------



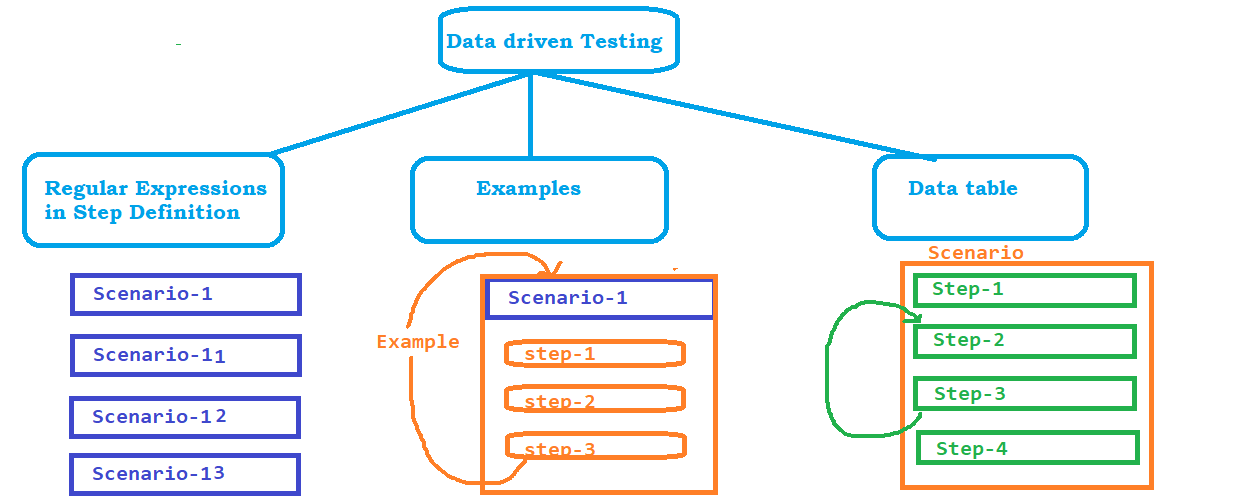
# Examples :

-------------



# DataTables:

-----------------



### DataTable Data structure:

====================

# asList : - List<String>

| Bhargav | Joshi | teacher | Grad School | Female | 3 | 01/01/2001 |

# asLists : - List<List<Sting>>

| Pavan | Joshi | QA | College | Male | 1 | 18/10/1997 |

| Rajat | Joshi | DE | High School | Male | 2 | 27/07/2000 |

| Bhargav | Joshi | teacher | Grad School | Female | 3 | 01/01/2001 |

# asMap : Map<String, String>

**| fname | lname | jtitle | edu | sex | exp | dob |**

| Pavan | Joshi | QA | College | Male | 1 | 18/10/1997 |

# asMaps: List< Map<String, String>>

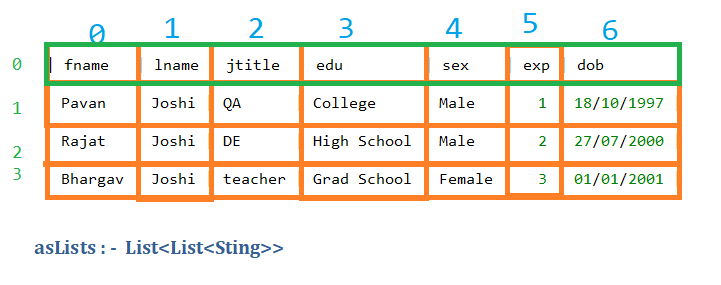
**| fname | lname | jtitle | edu | sex | exp | dob |**

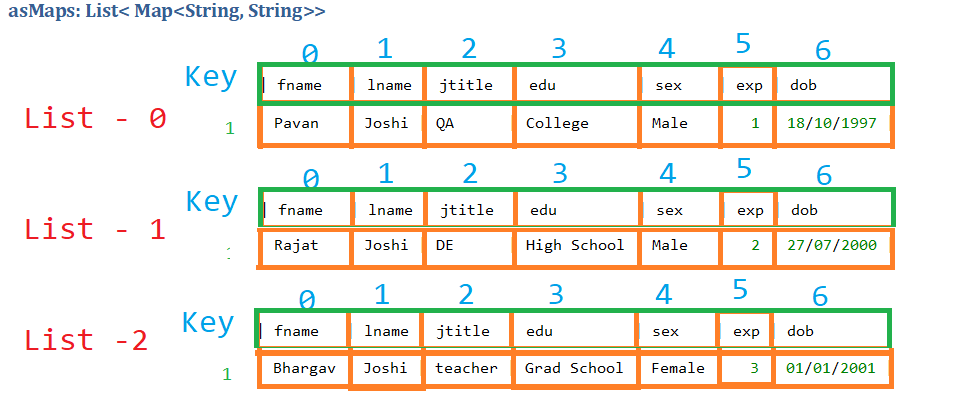
| Pavan | Joshi | QA | College | Male | 1 | 18/10/1997 |

| Rajat | Joshi | DE | High School | Male | 2 | 27/07/2000 |

| Bhargav | Joshi | teacher | Grad School | Female | 3 | 01/01/2001 |

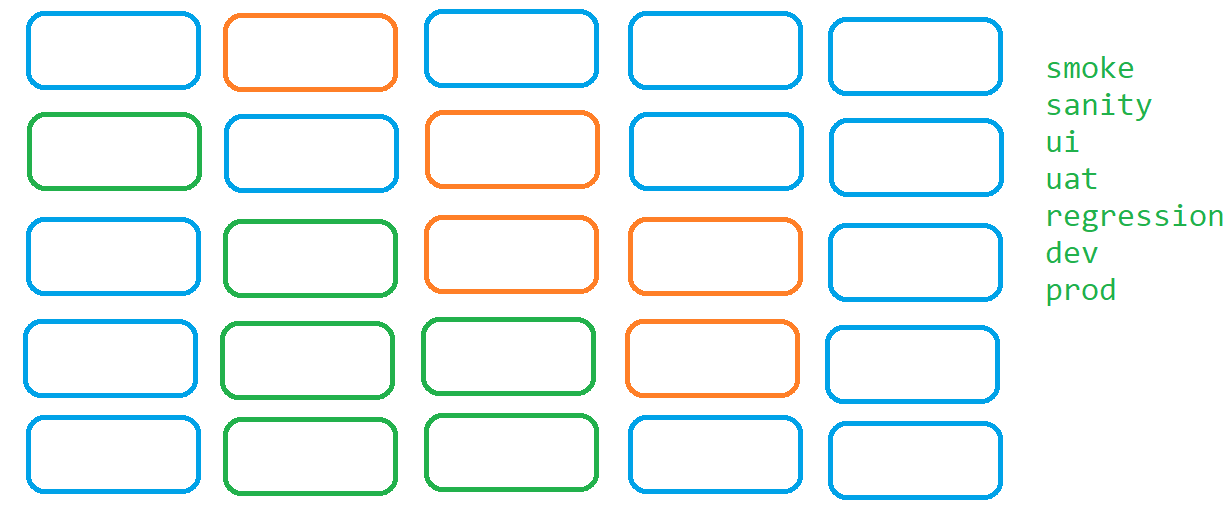
## Structure of Data

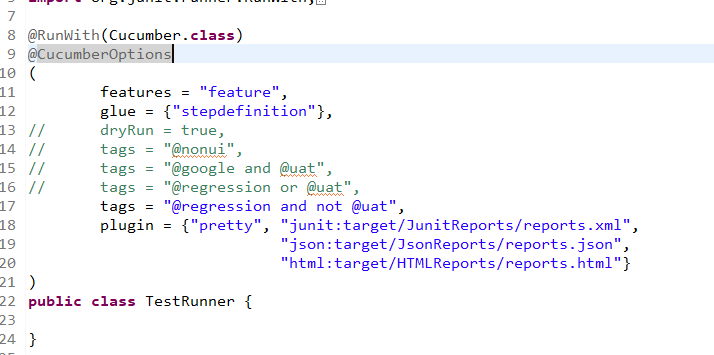




Tags

Tags are mainly used to group the tests and to control the test execution in cucumber.





Hooks

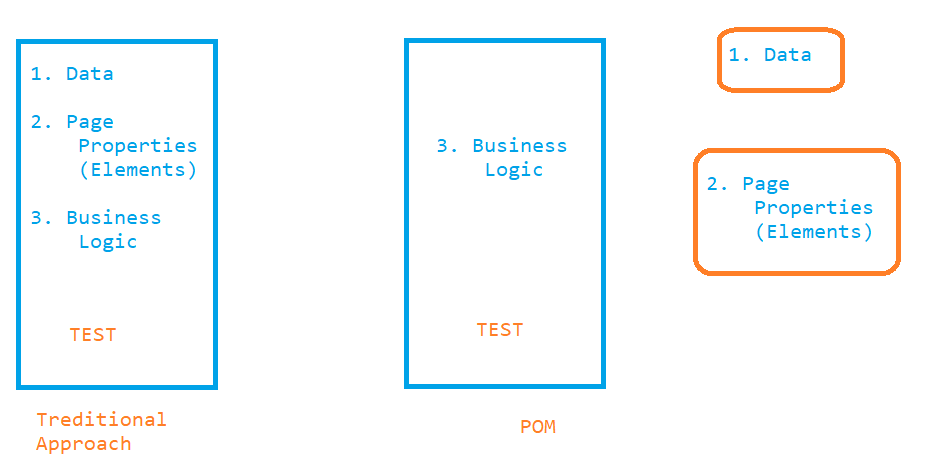
Hooks are the annotations which will be executed before and after executing

* Scenario
* Steps
* Tagged hooks
* ordered hook
* conditional hooks

We can even configure the Scenarios to execute for a specific tags

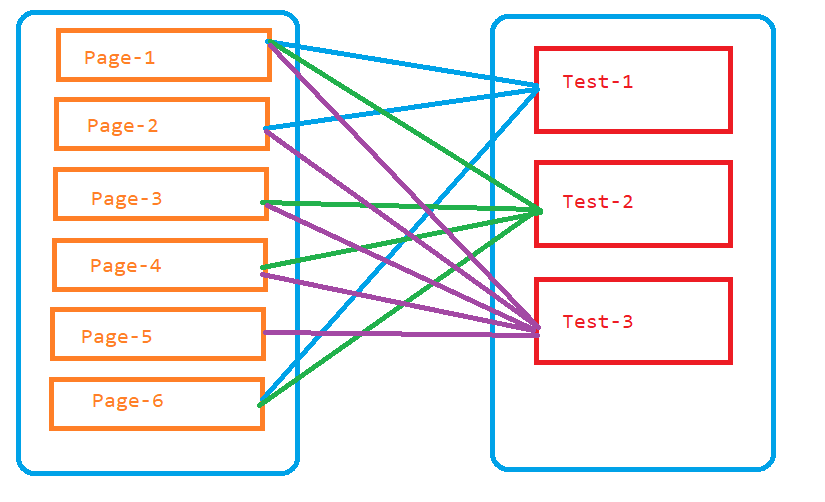
We can even configure the Scenarios to execute in the required order

Page Object Model: POM



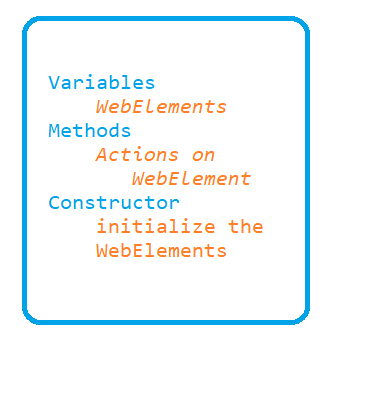
# Structure of POM

---------------------



# Component of PAGE :

------------------------



Creating Batch File

command line arguments - <https://cucumber.io/docs/cucumber/api/?lang=java>

