**But when we use frame work???? then --🡪 this** provide us

--- organized structure

---proper folder hierarchy

---- Separate place for separate code

----single execution point

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

**What is framework????**

**A framework is a pre-built set of tools and libraries which is used to develop software applications. Instead of starting from scratch, you can use the framework to speed up the process. It's designed to help you, organize your code, save time, and avoid common mistakes.**

**what is framework?**

**in English frame work mean structure, base of something, dhaancha, pinjra,**

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

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

**What is pom xml file??**

**This is configuration file which is considered as backbone of maven project. This file contains the dependencies, project information, plugins and repository.**

Then pom.xml file says that you just download the dependency I will automatically download all files.

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

**Project Information**: Contains basic details about your project, like its name, version, and description.

**Dependencies**: Lists the libraries your project needs to work. Maven will automatically download these from the internet.

**Build Configuration**: Defines how your project is built, like which tools and steps are used to compile and package the code.

**Repositories**: Specifies where Maven should look online to download the libraries or tools needed for your project.

**Project Structure**: Tells how the project is organized, like where the source code, test files, and other resources are stored.

**Plugins**: Adds extra tools to Maven for tasks like compiling code, running tests, or packaging your project into a file.

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

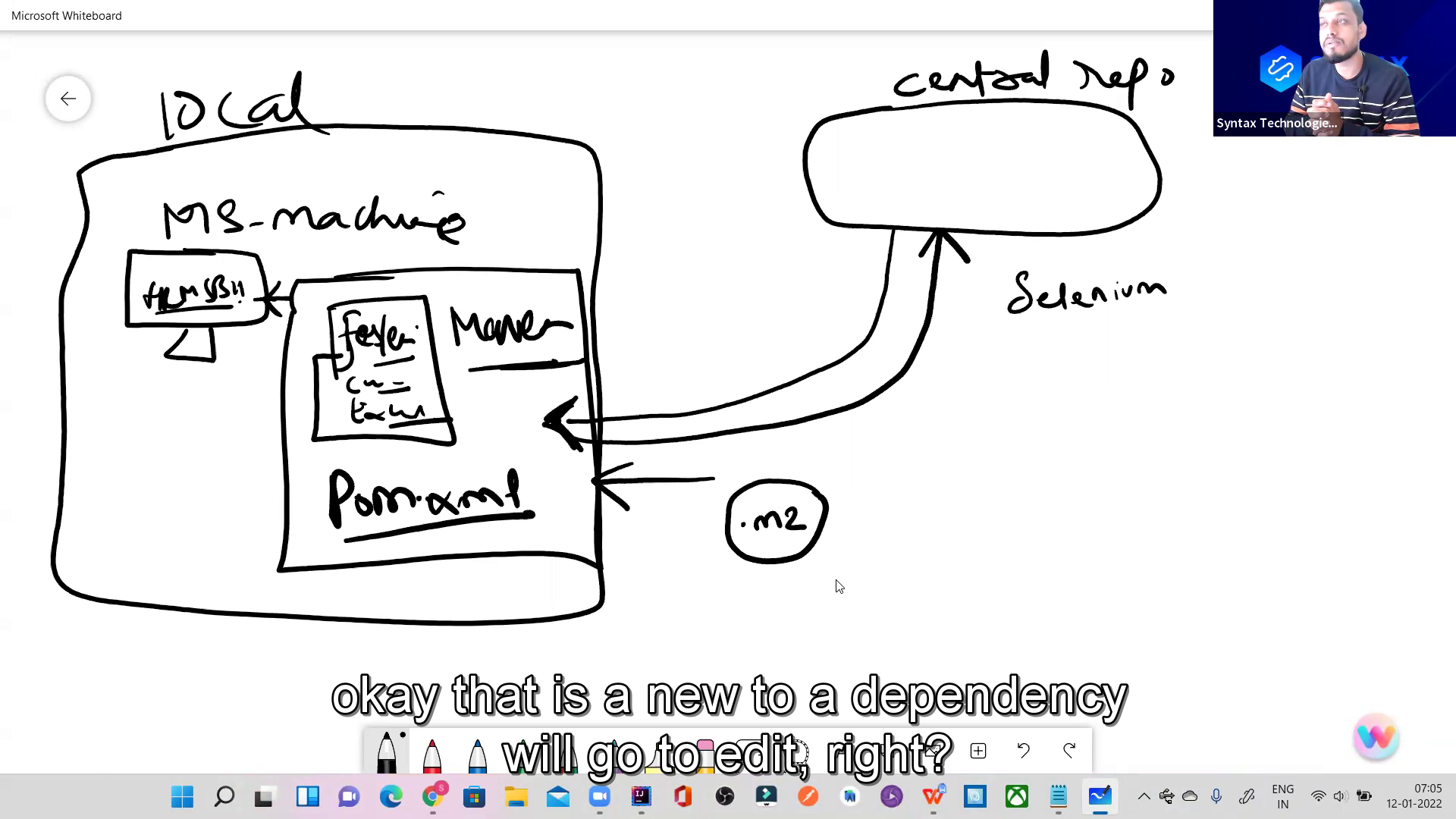
**How maven works, mean how it download jars and drivers automatically just by adding the dependency??**

When we put dependency in pom file this local pc is connected to central repository

Pom file send signal to repository is this dependency available here the central says yes and he transfer this file when we refresh our pom file. In our pc .m2 folder is created which store all jars and drivers in local pc.

**This is one scenario;**

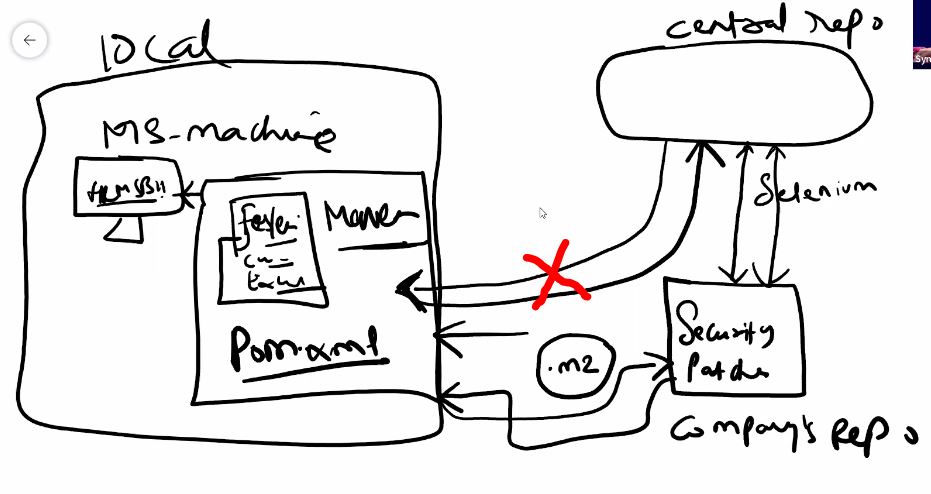
In this photo local machine or local repository is directly connected to central repository. On central repository is there are lot of adds, there is a highly security risk.



**2nd scenario;**

1. company repository

2.security patches.

****

**What is this mean?????**

<groupId>org.apache.poi</groupId>  
<artifactId>poi</artifactId>  
<version>4.1.2</version>

This dependency is present in pom file and this dependency pom file says to central repository , hey central do you have this group-id and this artifact-id and this version , if you have then please transfer this to me local machine, then in local machine .m2 folder is created and jars and libraries are downloaded in this folder.

Here are main two scenario or you say advantages and disadvantages of central repository.

Mariyam is project manager and five people are team member

Then everyone adding all jar files in his machine from central repository.

But here is disadvantage of this may be one member is downloading selenium version 4.2 of dependency or other person downloading other version if five people are downloading different, different version then some using updated version and some using older version then there might be problem or error, because someone using less version and some using new version then there may be error of whose code which is using older version. Because each version has different functionality.

What is solution of this case

In this case only Mariyam downloads jars and she will upload code on git hub and all other members will copy that code and will use same code and jars and drivers without no error.

2nd scenario if you are working on government project and they never allow you to direct link with central repository. Because of security issue. Then in this case company will add another repository inside company repository and this company repository will connect to central repository and this company repository will have security patches, and in this case your pc will connect indirectly with company repository to save from security crack issue.

Central repository mean maven repository maven repository website on internet or website of maven. It has lot of add on his page there may be security issue.

What is web Driver this is equal to Chrome driver this is an interface why this is interface because it has Own definition why it has own definition because it has Multiple situations why this has multiple situations because it has Multipl business cases

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

**What is jar and dependency???**

**JAR file**: It’s a pre-built Java classes and libraries your program needs to run. It often bundles libraries or complete applications.

**Dependencies**: These are external classes or libraries that are required to run your program but aren’t present in Java by default. For example, if you're using Selenium for automation, it would be a dependency for your program.

Example of Ammar who is business owner of automobile industry.

**What is cucumber?**

It is a testing framework software tool which test other software’s.

The Cucumber is a testing framework which is used for testing other software application.it supports BBD which means behavior of application is written in Gherkin language. Gherkin is simple plain English language, so that non-technical stakeholder can understand the application behavior.

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

**These test cases describe how a feature of your application should behave,** making it easy for both developers and non-technical people to understand.

**Why we need cucumber** when there are lot of tools like tesNg, selenium, java, maven.

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

**What does cucumber do ? benefits of cucumber.**

1. It increases the **Transparency**
2. Remove the **communication gap**
3. It engages more clients.
4. Cucumber solves the communication gap
5. **More collaboration** (kaam ma shirkat ) if ammar comes and see the project in gherkin language he will give suggestion or feed back and will collaborate more.
6. **Focus on end user**, it means the person who will use that software. A good tester consider himself as user of that software and use that software itself to check the functionality.
7. **Business value,** when you use BDD it mean this module provide the clients to tell their priority what page they need first and what things they need first to complete. When the priority pages are mage owner will start to do work will earn money and this will increase the business values.

Cucumber supports gherkin language

**What is gherkin language?**

Gherkin language is nothing just plane simple English language. Which we will write with some set of rules.

**What is BDD??**

**Behaviour of the test cases is written in gherkin language. Instead of showing the actual code showing the functionality of application to the stakeholder.** Behaviour is going to drive the framework.

**----------------------------------------------------**

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

If we are going working with gherkin language there are set of protocols which we are follow.

**What are those set of protocols?** There are some keywords these keywords has specific meanings.

Whenever we deal with gherkin language when ever we deal with BDD part we have to keep these keywords in mind. Which keyword what mean.

1. **Given** - --- > precondition ------ Given user is navigated to the HRMs application
2. **When** -- -🡪 actions -🡪 when the user enter password and username

like send keys, click, select dropdown, double click etc.

1. **And** ---- -> supporting tag, this tag comes with every other keyword for support.
2. And user click on login button
3. **Then** ---🡪 expected output, result ---🡪 Then user should be able to see dash board.
4. **Feature**-----🡪 this keyword describe the functionality or user story.
5. **Scenario**-------🡪 this keyword describe how many possibility are there to test the functionality.
6. **Background**--------🡪 this keyword takes all common steps present in all scenarios in one feature file.

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

**There are three types of frame work**

1. **Keyword driven framework** ----------🡪 generally base on keyword, we pass the data that’s why we called KDD, keyword is going to drive
2. **Data driven framework** ---------------🡪 when your script execute across multiple sets of data it is called DDT, data is going to drive/ test next generation is DDT framework.
3. **Behavioural driven development framework.** -----------------> instead of writing showing or writing the code you write the behaviour of application, behaviour is going to drive. Now what does mean of behaviour, it is to define some one’s nature is called behaviour. What it do, what it say, what it eat, what it like.

**Whose behaviour we are defining?** we are defining the behaviour of test cases which we write multiple test case. It means We are writing in plain English language

**BDD** ---- defines the behaviour and to define the behaviour the language which we used is gherkin language.

BBD------ >> in simple words, instead of showing actual code to the clients, show him the behaviour of application/show him the functionality of the application/show him the user story. Behaviour is going to drive the framework.

First we will write test cases in gherkin language and then we see how to handle them.

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

**What is hybrid frame work?** if frame work supports both DDT and BDD then it is called hybrid frame work.

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

**DDT :** in ddt mean there is data code, there is my code, if my execution automation scripts execute across multiple sets of data this is called data driven test.

**BDD:**  it simply means it define the behaviour of test case, here is plain simple English language you will find. If there is my plain English language,

the scenario in which I am writing these scenarios also executing across multiple sets of data then we say it is DDT and BDD both mean it is hybrid frame work.

**BDD defines the behaviour and the language which defines the behaviour is gherkin language.**



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

**Class 02 Date; 17-12-2023**

Feature --------one functionality is called one feature. describes the overall feature of functionality being tested. The file in which we write all the feature or test cases or scenario is called feature file.

Scenario--------test case is called scenario. One test case, one scenario. Describes a specific test scenario or case related to the feature.

Feature --------- > User Authentication

Scenario --------- > Successful Login

Given --------- > the user is on the login page

When --------- > the user enters valid credentials

Then --------- > the user should be logged in successfully

Scenario --------- > Unsuccessful Login

Given --------- >the user is on the login page

Then --------- >the user should see an error message

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

Feature is what the functionality you want to test ????? example; user story 123 search employe.

Scenario ; how to test the functionality. Example ; search employee by id.

**In frame work we need**

1. java----------------- > jdk

2. cucumber --- > dependency

**i.  [Cucumber JVM: Java](https://mvnrepository.com/artifact/io.cucumber/cucumber-java)**

[**io.cucumber**](https://mvnrepository.com/artifact/io.cucumber)**»**[**cucumber-java**](https://mvnrepository.com/artifact/io.cucumber/cucumber-java)

**ii.  [Cucumber JVM: JUnit](https://mvnrepository.com/artifact/io.cucumber/cucumber-junit)**

[**io.cucumber**](https://mvnrepository.com/artifact/io.cucumber)**»**[**cucumber-junit**](https://mvnrepository.com/artifact/io.cucumber/cucumber-junit)

**3.selenium**

i**.**[**Selenium Java**](https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java)

[**org.seleniumhq.selenium**](https://mvnrepository.com/artifact/org.seleniumhq.selenium)**»**[**selenium-java**](https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java)

**ii.  [Selenium Chrome Driver](https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-chrome-driver)**

[**org.seleniumhq.selenium**](https://mvnrepository.com/artifact/org.seleniumhq.selenium)**»**[**selenium-chrome-driver**](https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-chrome-driver)

**4.excel reader jars/or library**

i.**.**[**Apache POI Common**](https://mvnrepository.com/artifact/org.apache.poi/poi)

[**org.apache.poi**](https://mvnrepository.com/artifact/org.apache.poi)**»**[**poi**](https://mvnrepository.com/artifact/org.apache.poi/poi)

## . [Apache POI](https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml-schemas)

**5.webdrivermanager**

[WebDriverManager](https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager)

[io.github.bonigarcia](https://mvnrepository.com/artifact/io.github.bonigarcia) » [webdrivermanager](https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager)

[**org.apache.poi**](https://mvnrepository.com/artifact/org.apache.poi)**»**[**poi-ooxml-schemas**](https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml-schemas)

**in intellij download plugins**

**1. cucumber for java**

2.gherkin

3.properties

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

**What is step declaration??**

Step declaration; writing the cucumber scenario in feature file in gherkin language is called step declaration. Only feature file you cannot execute. Step declaration is only skeleton.

Scenario: User logs into HRMS

Given user navigated to hrms website

When user enters valid credentials

Then user should be logged into HRMS

**What is step definition??**

**Step definition;** writing the java code for step declaration is called step definition. a step definition is a piece of code associated with a specific Gherkin step in a scenario.

**Annotation (@Then):** This indicates that the following method is a step definition for a Cucumber scenario's "Then" step.

This is a step definition of a cucumber’s scenario.

@Given ("user navigated to hrms website")  
public void user\_navigated\_to\_hrms\_website() {  
 openBrowser();  
}

**Gherkin is a language used to write human-readable descriptions of software behaviours**.

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

Class 03 date;12-20-2023

**Benefits of cucumber?**

Why we need cucumber if there is testNG??

1. increase the transparency, with clients and software team

2. reduce communication gap

3.cucumber supports BDD/gherkin language, it is easy to understand for non-technical people.

4.focus on end user – you always say user user mean you are using this app as user.

5. you don’t need to write manual test cases; you can link your jira with framework.

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

We need a feature file to write test scenario in our cucumber frame work.

In a feature file we can write lot of scenarios, like login with valid, login with invalid and other login

But in a feature file you can have only one feature. Because in one feature file you are describing one feature of any software. This feature may be very large. One file describes only one user story

Every step declared in cucumber scenario generate the java code.

And keyword is supporting key word. In step definition and keyword is not annotation, the annotation is only showed the actual keyword for which this is using for supporting.

**Multiple inheritance in not allowed in java.**

1. login step definition are present in one class login class

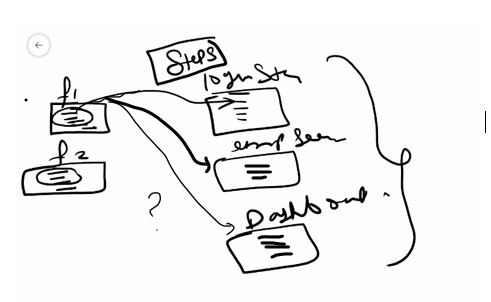
2. login step definition you don’t need to copy and paste in employee id search class. If you do this there will be multiple inheritance.

Suppose you have multiple classes in one java class, and if there comes error it is difficult to solve that error rather in if they are separately created.

Maven knows which class to use as the entry point.

**How multiple inheritance not allowed????????**





In above f1 feature file is confused that from which class I have to take step definition of java code. This is multiple inheritance not allowed in java.

Step definition; to implement the code for steps written in feature file. We use step definition.

**What is Runner class??**

Runner class is **the configuration class** it is considered as back-bone .Runner class is used to execute more than one feature files. It works as a bridge between your feature files and step definitions.

**Configuration:** It's considered the "backbone" because it contains crucial configuration settings that determine how your tests will run.

**Annotations:** In Cucumber-JUnit, the Runner class is typically annotated with @RunWith(Cucumber.class) and @CucumberOptions.

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

**What is Junit???**

JUnit is a popular testing framework for Java that is used to write and run unit tests. Unit testing is a software testing technique in which individual units or components of a software application are tested in isolation to ensure they behave as expected.

**Junit gives methods, assertions, runner class, annotations**

**Annotations**: JUnit uses annotations to define and manage test methods and setup. Some common annotations include:

* @Test: Marks a method as a test method.
* @Before: Runs before each test method.
* @After: Runs after each test method.

**Assertions**: These are methods used to check if the test results are as expected. For example:

* assertEquals(expected, actual): Checks if two values are equal.
* assertTrue(condition): Checks if a condition is true.
* assertFalse(condition): Checks if a condition is false.

**Annotations:**

**Is just like a label, which mean what test piece or which test part or which test step do you want to test.**

**What is unit testing???????**

Unit testing is a software testing technique where individual units or components of a software application are tested in isolation. The term "unit" refers to the smallest testable part of an application, usually a function, method, or class.

**We are using cucumber with junit.**

import org.junit.runner.RunWith;  
  
@RunWith(Cucumber.class) // this is linking cucumber and junit.

**@RunWith(Cucumber.class)**: This line is using the **@RunWith** annotation to tell JUnit to use the Cucumber test runner (**Cucumber.class**). This is a key part of integrating Cucumber with JUnit. It essentially says, "Hey JUnit, when you run tests in this class, use the Cucumber test runner to interpret and execute the Cucumber features."

**We are going to configure a runner class.**

The purpose of runner class is to execute all feature files not a single feature file from one point.

@CucumberOptions ()

The **@CucumberOptions** annotation in Cucumber is used to provide various configuration options for your Cucumber tests. It allows you to specify settings such as the location of your feature files, the location of your step definitions, formatting options for the test results, and more.

**There are 6 cucumber options generally used**.

1. **feature** ------is cucumber option where you specify the path of feature files present in your project.
2. **glue**---------- is a cucumber option which is used to make link between feature file and step definition
3. **dry-run** ------is a cucumber option which is used to generate unimplemented steps, and saves your time by just scanning the code.
4. **monochrome** ---------is a cucumber option which is used to remove unnecessary code from console
5. **plugins---------------------** is cucumber option which is used to generate html and json report for your test cases.

**plugin = {"pretty", "html: target/cucumber.html"}**

**plugin = {"pretty", "json: target/myFile.json"}**

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

**What is background keyword??**

Background keyword is used to write all the steps which are common in all scenarios present in a feature file. . You can say that this is precondition before run of every scenario.

These steps are executed before each scenario in the feature file.

Common steps mean common between 2 or more than 2 scenarios not in a single scenario.

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

**Class 04** Date;12-23-**2023**

**What are hooks???**

The steps which are common in all the feature files like opening browser and close browser written in a class called hooks. There are two hooks in cucumber @before and @after. We provide hooks for precondition and post condition to write common steps of all feature files.

We provide these hooks under the step package where our java code is present.

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

**There are two hooks in cucumber**.

**@Before** hooks will be run before the first step of each scenario.

**@After** hooks will be run after the last step of each scenario, even when there are failing, undefined, pending or skipped steps.

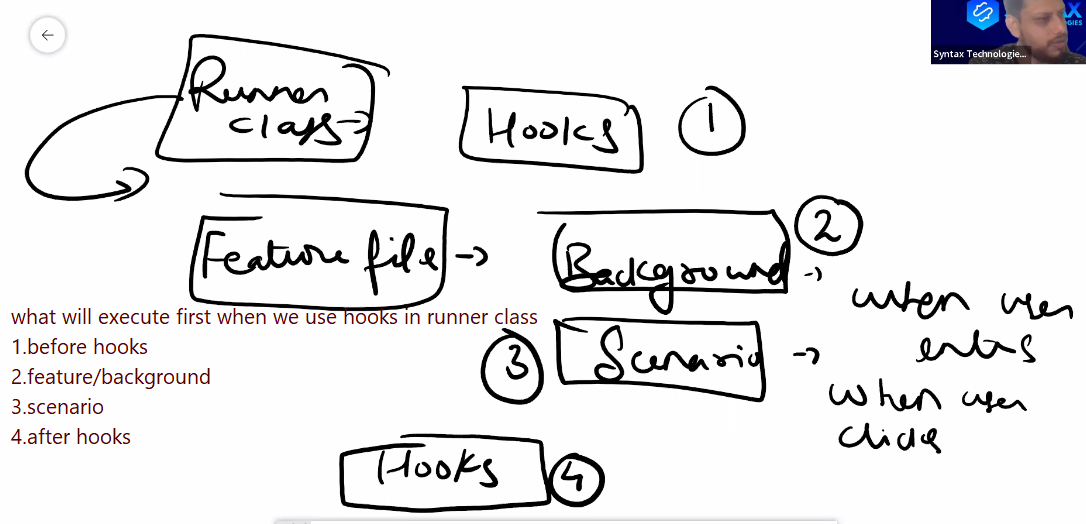
We just need to define hooks, no need to associate the hooks, cucumber takes care of associating.

 @**Before Hook**: Runs before each scenario.

 @**After Hook**: Runs after each scenario.

 @**BeforeStep Hook**: Runs before each step.

 @AfterStep **Hook**: Runs after each step.



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

**There could be many runners class**

**1. Smoke class**

**2. Regression class**

**-------------------------------------------------------------------------------------------------------------------**

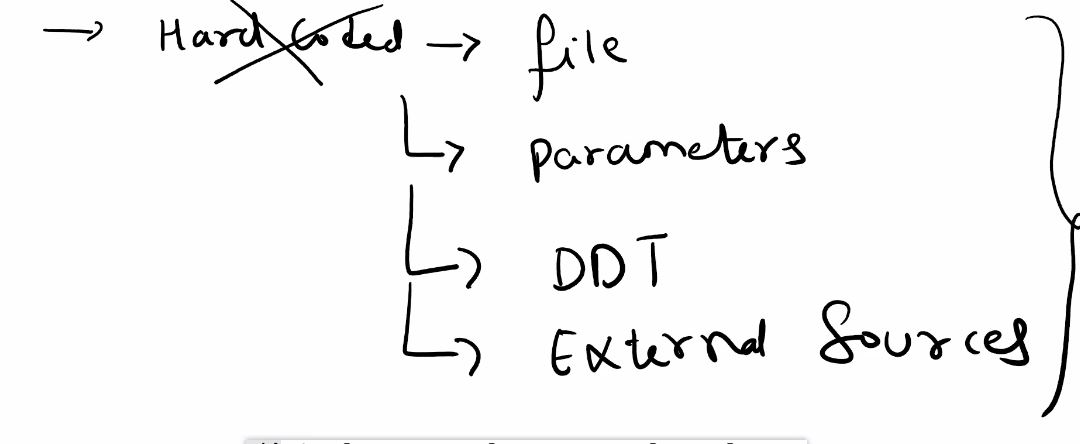
**There are many ways to pass the data in test cases.**

* 1. **Passing hard coded data -- in step definition directly**
  2. **Passing data from properties file ---use configReader class password and username**
  3. **Passing data from excel file**
  4. **Passing data parameter like data table and example table**

**------------------------------**

**What is Data Driven Testing???**

**You can pass multiple sets of data in a single scenario this is called data driven testing.**



**1. Hardcoded Data (Not recommended):**

* This involves directly placing the data in the step definitions or feature files

**2. Files:**

* You can read data from external files such as **property file**, or **Excel** files

**3. Parameters: from feature file**

**4. Data-Driven Testing (DDT):**

* + **Examples Table** in Gherkin: You can define multiple sets of data directly in the feature file using the Examples keyword.
  + **@DataTable** in Cucumber: This allows structured data to be passed directly in feature files using tables.

**-------------------------------------------------------------**

**Scenario Outline:**

* A **Scenario Outline** is a keyword used with Examples table to pass multiple sets of data to test single scenario.
* A **Scenario Outline** is typically paired with an Examples table that provides the data.

**Example:**

Example table is applicable for whole scenario.

There is no need of loop to iterate across multiple sets of data , examples table automatically iterate the scenario until the condition is met.

Every time open the browser and close the browser.

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

Passing data using placeholder directly in scenario using double quotes.

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

**What is data table??**

The Data Table class is used to pass multiple sets of data (typically in tabular format) to test a single scenario.

In Cucumber, **Data Table** is a class **provided by Cucumber itself**, not JUnit or Maven.

Data table is applicable only for one step in scenario.

**What is Difference between data table and examples table??**

* Data table is declared with scenario keyword and examples table is declared with scenario outline keyword
* Data table not need examples keyword
* Data table is applicable only for one step
* Examples table is applicable for entire scenario
* Data table open and close browser just once.
* Examples table open and close browser for each iteration
* Example tables waste lot of time to unnecessarily open and close browser after each iteration.

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

**What is difference in scenario and scenario outline???**

**Scenario is a keyword which mean how many possibilities are there to test the user story or functionality.**

**Scenario Outline:**

* A **Scenario Outline** is a keyword used with Examples table to pass multiple sets of data to test single scenario.
* A **Scenario Outline** is typically paired with an Examples table that provides the data.

**Example:**

Example table is applicable for whole scenario.

There is no need of loop to iterate across multiple sets of data , examples table automatically iterate the scenario until the condition is met.

Every time open the browser and close the browser.

Class 5 is just excel file reading code

**Class 06**

What is Pom??

**What is page object model.?**

It is a design pattern; it enhances the test maintenance and reduce/prevent the duplication of code.

It is going to help us to make lot of things reuseable.

We divide our project into four packages

1. Base class
2. Common utilities
3. Your pages
4. Your test

**What is page factory and without page factory ?**

This is a class in selenium library.

Both are approaches in selenium how to implement the page object model POM .

Main difference between them is how the elements on web page are initialized and accessed.

* Page Factory approach uses annotations (**@FindBy**) to declare WebElements, and these elements are initialized using **initElements()** method of PageFactory class.
* Non-Page Factory approach initializes elements using **driver.findElement()** method directly within the Page Object class without any annotations.

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

**What is maven lifecycle????**

Take screen shot. To generate screen shot run the test from pom file.

In pom file you have to write the name of runner class and in runner you have to write the tags which you want to run like login , dash board.

You can also run you code wihtou intellj in command prompt

Mvn clean --- to clean the project

Mvn test -------to run the code

**What is maven life cycle???**

**The Maven lifecycle is a series of steps that Maven uses to build, test, and make a package of software project.**

**What is benefit of maven life cycle??**

It automates repetitive tasks like compiling code, running tests, and packaging applications, saving developers time and effort.

**What is mean of build???**

**"build"** refers to the process of transforming source code into a runnable application or library. This typically includes several steps:

**Clean**—clean the target folder junk files from the project

**Verify**—verify the project is error free

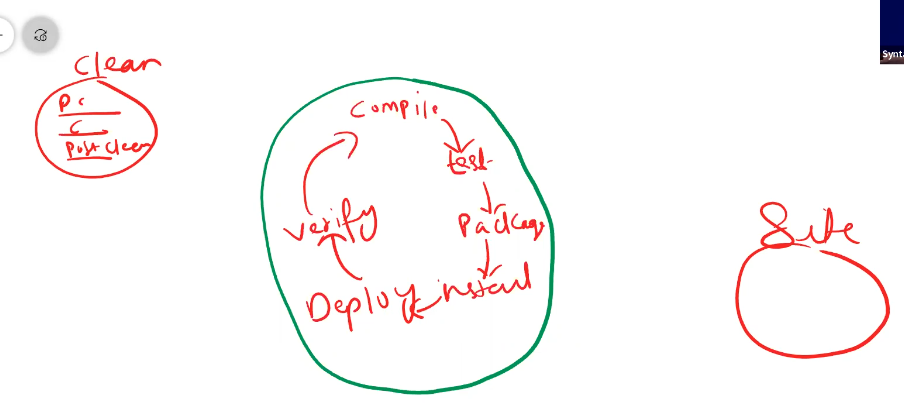
**Compile** --- convert source code into byte code

**Test**---run the project to check every thing is runnable

**Install**—install if some jar or dependency is missing

**Package**—make the project as a package or library

Deploy--- deploy to the server.



What is sure fire plugin???

**Parallel Test Execution**: It allows running tests in parallel to speed up the test execution time.

**Reports**: After running the tests, Surefire generates reports summarizing the results (pass, fail, skipped tests) in formats like text, XML, and HTML.

**What is maven??? What is build, what is POM file contains???**

Maven is a build management tool, build mean it compile the source code, execute the test and make a package of compiled code into a deployable format like jar or war file.

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

**Build**: In simple terms, a build is the process of converting **source code** into a runnable software application. It includes steps like compilation, linking, and packaging the application for deployment.

**POM File (Project Object Model)**: Maven uses an XML file called pom.xml to define the project's configuration. This file contains:

* Dependencies (external libraries your project needs)
* Build configurations (how to compile and package the project)
* Plugins (for additional tasks like generating reports, running tests)
* Project details (name, version, etc.)

**Folder Hierarchy**: Maven enforces a **standard project structure**. For example:

* **src/main/java**: Your application's source code
* **src/test/java**: Test code
* **target**: The directory where the compiled and packaged application will be stored