

## \* Page Object Module (POM)

PAGE NO.	
DATE	16/6/21

- \* POM stands for page Object Module.
- It is a java design pattern used to design a classes in test script. for that purpose we need to maintain all the webelement (which we hv to handle) into seperate class.
- POM class follows encapsulation concept in any project.
- As we all of you know that encapsulation is one of the pillar of OOPS.
- So, Basically encapsulat" is the process of binding data & corrusponding methods together into a single unit.
- In that we can declare data mem. (variable) as private & method as public . and
- used getter & setter method to read / write data.
- so same thing we are doing into POM class.
- Data member should be declared globally with access level as private . & method as public.
- And that data member (variables) are Initialized within a constructor.
- In POM, no. of data member will depends on no. of components that we have to handle on webpage.
- POM class is used in all test automation that create different classes to used diff". obj for handling webelements . like POM class & test class .
- POM class will not contain main method that's why we never execute POM class . so for that we need to declare another class i.e Testclass.

④ = annotation.

PAGE No.	
DATE	/ / /

- So, Test class will contain all navigation steps to test an application by using POM class.

#### \* Advantages of POM

- ↳ - POM is design pattern which helps us to build our framework strongly
- It reduces the code duplication.
- Improve test maintainance.
- Script will be more readable.

#### \* Disadvantage -

- ↳ The big disadvantage of POM is,  
POM will initialize data members before performing action on it.

It means, sometime application may contain some components/webelements which will be hidden or will not be displayed. so we can inspect that element but we are not able to perform action directly on that hidden webelement.

So, Overcome this problem, we have an extension of POM class i.e. Page Factory.

#### \* Page Factory -

- Page Factory is class which is inbuilt in POM concept.
- It is class, which contain static method i.e 'init()'.
- To Initialize datamembers in a page factory we have method i.e 'initElement()' with in constructor
- initElement() - will be Initialize data members by identifying each component those are present in webpage by using @FindBy annotation, which takes locator type as an argument.

`@findby` → used to findout live webelement, non-live webelement, hidden element.

**Syntax -** `Pagefactory.initElement (webdriver ref, this)`

- while executing testscript initElement method will convert all the data members into `@FindBy` annotation

`@findBy (xpath = " ")`

this process is known as basic initialization / early initialization.

*Imp  
Interview \**

## Maven (Apache - Maven)

- The Maven project is hosted by Apache s/w foundation where it was began with Jakarta project
- Maven is a build automation tool used for Java project based on the concept of a POM class.
- Maven can also used to build and manage projects written in many languages like C#, Ruby, Scala and many more.
- Maven is used to define project structure, dependencies, build, test management.
- Maven projects are configured using a POM, which is stored in a `pom.xml` file
- using `pom.xml` you can configure dependencies needed for building testing & running code.

- Maven automatically downloads the necessary files from the repository while building the project. there is no need to import external jars.

`pom.xml`  
file

Maven project - includes project & configuration info such as dependency , source directory , construct directory , goals , plugins etc. used by Maven to build the project .

#### \* History of Maven -

- Maven is a "Yiddish word ". which is created by 'Jason van zyl' in 2004 at Apache slw foundation.
- The first version of Maven is 1.0 and latest version of Maven is 3.8.1 (2021).

#### NOTE-

Maven is project management tool & project build tool  
It is used to check the compilation issues betn framework components whenever multiple test engine integrates their files into same framework.

- If no compilation issues in the framework then It will provide build success msg. else build failure

#### \* Installation of Maven -

go to Eclipse → click on Help → eclipse market place → type in search bar i.e maven Integration for eclipse's

#### \* Create Maven Project -

open eclipse → file → new → project → select Maven project  
→ Next → Next → enter groupId → ArtifactId  
→ finish .

**Info**

## Difference between HTML & XML

- \* HTML - stands for Hypertext Markup language.
  - HTML is presentation language which is used to display data and focuses on how data looks.
  - HTML is not case sensitive
  - HTML has its own predefined tags.
  - In HTML, it is not necessary to use a closing tags.
  - HTML is static becoz it is used to display data.
  - It does not preserve whitespaces.
  
- \* XML - stands for Extensible Markup language.
  - XML language is platform independent which is used to store & transport data.
  - XML provides a framework to define markup language.
  - XML is case sensitive.
  - XML is neither a presentation language nor a programming language.
  - You can define tags according to your need.
  - It is dynamic becoz is used to transport data.
  - XML preserve whitespaces.

NOTE - Before writing code, we need to add dependency i.e Apache poi from MVN repository.

Info

## \* Maven POM (pom.xml)

PAGE No.	/ / /
DATE	/ / /

- \* Maven projects are configured using a POM which is stored in pom.xml file

Project Home

src

main

Java

resources

Test

Java

resources

Target

Pom.xml

\* project home directory contains the pom.xml.

(Dir. Directory structure of maven project.)

{ src/main/java → contains the deliverable Java sourcecode for the project.

src/main/resource → resources for the project, such as property files.

src/test/java → contains the testing Java sourcecode (e.g JUnit 4 TestNG)

src/test/resource → resources for testing.

Interview

\* What Is POM (Maven) ? [Project Object Module]

L - POM is Fundamental Unit of work in Maven.

- It is a xml (pom.xml) file that contains information -

\* By default maven will take care of JUnit dependency. becoz, it is used to test our code like most

about the project and configuration details used by Maven to build the project.

- It contains default values such as target which is build directory , src/main/java - which is source directory & the test source directory i.e src/test/java and so on.
- POM also contains the goal & plugins . while executing a task / goal , Maven looks for the pom in the current directory .
- It reads the POM , get the config. info. & then executes the goal .
- some of the configuration that can be specified in the POM such as
  - project dependencies
  - plugin
  - goals
  - build profile
  - version and so on .
- Before creating a pom.xml , we should decide first the project - groupId , artifactId , & version .
- becoz. all Pom.xml files require the project element i.e groupId , artifactId , version etc. (Mandatory ) these 3 fields are mandatory becoz, they help to id uniquely identify the project in repository .
- groupId - this is an Id of project's group.
- artifactId - this is an Id of project (artifact)
- Version - the version of project under specified group.
- also Pom.xml file contain proj. element- like project , dependency , scope etc , modelversion etc.
- project - It is root element of .xml file
- ModelVersion - specifies the version of model.

- dependencies - defines dependencies for the project  
The dependencies are gathered in the pom. when you run a build or execute a maven goal, these dependencies are resolved & then loaded from repository.
- dependency - In Maven, dependency is another archive i.e JAR, ZIP and so on. which your current project needs in order to compile, build, test & run.
- It is used inside dependencies.
- scope - defines scope for maven project, it can be compile, runtime, provided, test & system.
- plugin - The maven plugins are central part of maven project , it is used to perform specific goals.

### \* Advantage of Maven -

- 1) No need to add jar files in each project.
- 2) Creates tight directory structure .
- 3) Build and deploys the project.
- 4) Maven plugins are reusable.

### \* Build lifecycle - It can be used to give order to goal execution like @ what are the build phases ?

- test
  - install
  - deploy.
- 1) validate
  - 2) compile
  - 3) test
  - 4) deploy

### \* What is repository ?

→ It is a directory where all the pom.xml files are stored.

@ archetype ? → It is maven plugin. It creates the project structure

PAGE NO.	/ /
DATE	/ /

Build Life Cycle

Maven is based on the concept of build lifecycle there are three built-in build lifecycles i.e default, clean, site.

- default → default lifecycle handles your project deployment

clean → handles project cleaning

site → creation of your project's site documentation

V.Imp

② How to add dependency with maven repository ?

L → - Select pom.xml file from project & open pom.xml and after that just go to google and type MVN Repository into search box. then

- open then MVN Repository window which contains the multiple category of web testing .

- But I've work on selenium java that's why search selenium -java and we will see selenium java result. just click on that & select most usage version of selenium i.e 3.141.59 .

- Click on 3.141.59 and we will see central repository and copy the dependency from maven .

- Again open go to pom.xml and paste dependency into dependencies .

- You can add multiple dependency into (under) dependencies in this way . And run pom.xml and we will see dependency will added in Maven dependencies .