MAVEN

1. **What is Maven?**

- A build tool and command prompt tool that called POM xml file that calls my runner class and manages my dependencies.
- Maven is a build automation tool or a project management tool. With Maven we can import all libraries and can also create project structures. In Maven we have many inbuilt templates. These templates are called archetypes. A Maven is basically a tool used to compile our applications.
- Command Prompt mvn archetype; generate
  - Creates project
- Choose a # press enter
- Choose a # press enter
- GroupId; com.nameOfProject (usually a reversed domain name, like com.example.foo)
- ArtifactID; testmavenproject
    - Version enter
    - Package enter
    - Y; enter

2. **Why Maven? How it helps you developing your project effectively?**

- It helps to develop and managing project structure or applications like deployment, clean, packaging, jar and many more features for the Java-based project.
- In another word, it is a Java tool. If you want to create a sample project or skeleton project you can use Maven. It is an automated build tool. The Maven focused on simplicity that it generates intelligent starters and assumes intelligence defaults. It also covers build-oriented phases in Application Lifecycle Management i.e. testing, deployment, builds management, and release versioning.
- **It helps** to setup project very quickly and it avoids complicated build files like build.xml. Maven required files like POM.xml; it serves the purpose for Maven only. POM.xml is a collection of dependencies of your Java Project which one can specify to Maven and then Maven will download all of them from the internet and then store it to some repository i.e. local repository, central repository, and remote repository.

3. **What is Maven Artifact?**

- An artifact is a file, usually a JAR, that gets deployed to a Maven repository.
- A Maven build produces one or more artifacts, such as a compiled JAR and a "sources" JAR.
- Each artifact has a group ID (usually a reversed domain name, like com.example.foo), an artifact ID (just a name), and a version string. The three together uniquely identify the artifact. Example:

```
<groupId>org.seleniumhq.selenium</groupId>
<artifactId>seleniumjava</artifactId>
<version>3.11.0</version>
```

- A project's dependencies are specified as artifacts.

4. **Explain me the maven lifecycle?**

- Commands can only run in the same directory where the specific **pom xml** file is located
- 3 built in build lifecycles
  - <u>Default</u> → Handles your project deployment
  - <u>Clean</u> → Handles project cleaning
  - <u>Site</u> → Handles creation of project's site documentation

5. **A build lifecycle is made up of phases**
    - Validate → Validate the project is correct and all necessary information is available
    - Compile → Run the source code of the project (checking if there is any error or not, if not→ build success)
      → Target folder is created, and Reports will be stored here
    - Test
        - Test the compiled source code using a suitable unit testing framework.
        - Should not require the code to be packaged or deployed
        - Mvn D(VariableName) = testname → Run specific tests based on the parameter
    - Package→ Take the compiled code and package it in a distributed format, like JAR
    - Verify → Runs any checks on results of integration tests to ensure quality criteria are met
    - Install → Install the package into local repo, for use as dependency
    - Deploy → Done in the build environment, copies the final package to the remote repository for sharing with other devs and projects

6. **How do you convert maven project to eclipse project?**
    - Mvn eclipse

7. **How java projects are made?**
    1. Create folders/packages
    2. Add libraries/dependencies
    3. Create class files
    4. Compile
    5. Run tests
    6. Deploy

8. **Where do you find your dependencies/libraries?**
    - Mvnrepository.com
    - Update project if maven not working
        - When you have dependencies inside your pom file and you use update, maven will pull the JAR files from internet and add it to your project

9. **What is .m2 folder?**
    - Where your jar files/repositories are saved in your computer

10. **What is POM xml file?**
    - A file that manages the whole project
    - When you run a maven command, everything should be done through the pom.xml

11. **Versions of tools?**
    - RestAssured 3.3.0 release date: 2019-01-11

12. **Log4j?**
    - Used by any application
    - Example: LOG4J2 → From Apache
    - Records activity
    - Dev will look at the logs, look at the time, go to the IP address and see what going on if there was a bug
    - Loggers are very important part of applications and it keeps each step/event happened with timestamp
    - Normally logs are written programmatically into .log file
    - There are ready tools/libraries to add to any framework or application.
    - In java, the most famous logging library/framework is LOG4J from apache

13. **Purpose of logs?**
    - Help us debug the issues that you may have with application.
    - Sometimes when a bug is found in application, developers firstly check the logs. In order to see which steps the user was taking and application did not behave as expected.
    - Logs MAY help you find the source of the problem (in application perspective, not testing)

14. **What is the role of logs in Test automation?**
    - ● We are looking at console or html report to see the status of our test runs. If anything fails, we find from there.
    - ● If we implement logging into our framework, it will be another way of looking at automation execution steps and will help us find the problem whenever our test fails