

## 18/03/2025 - TASKS

### Task Questions:

- Maven life cycle
- What is pom.xml file and why we use it
- How dependencies work?
- Check the maven repository.
- How all modules build using maven
- Can we build specific module?
- Role of ui.apps and ui.content and ui.frontend folder?
- Why we are using run mode?
- What is publish env?
- Why we are using dispatcher?
- From where can access the crx/de?

### Task Questions and Explanations:

#### 1. Maven Life Cycle

- Maven follows a predefined lifecycle to build and manage projects efficiently. The key phases include:
  - **Validate:** Ensures the project structure and necessary information are correct.
  - **Compile:** Converts source code into Java bytecode.
  - **Test:** Runs unit tests to verify functionality.
  - **Package:** Bundles compiled code into a JAR or WAR file.
  - **Verify:** Checks the integrity and quality of the package.
  - **Install:** Stores the package in the local Maven repository for reuse.
  - **Deploy:** Uploads the package to a remote repository for sharing.

## 2. What is pom.xml file and why we use it?

- The pom.xml (Project Object Model) file is the core of a Maven project. It contains metadata, dependencies, build configurations, and plugins required to manage the project.
- It allows automation of builds, dependency management, and project configurations.

## 3. How dependencies work?

- Dependencies are external libraries required by the project.
- They are defined in the pom.xml file under the <dependencies> section.
- Maven automatically resolves and downloads them from repositories.

## 4. Check the Maven Repository.

- Maven repositories store dependencies and artifacts. There are three types:
  - **Local Repository:** Located at ~/.m2/repository.
  - **Central Repository:** Default remote repository (<https://repo.maven.apache.org/maven2/>).
  - **Remote Repository:** Custom repositories used by teams or organizations.

## 5. How all modules build using Maven?

- Maven supports multi-module projects where modules can be built together.
- The parent pom.xml file contains references to sub-modules.
- Run mvn install at the root level to build all modules.

## 6. Can we build a specific module?

- Yes, using the command: mvn install -pl module-name -am.
- This builds only the specified module along with its dependencies.

## 7. Role of ui.apps, ui.content, and ui.frontend folder?

- These folders are used in Adobe Experience Manager (AEM) projects:
  - **ui.apps:** Contains application-specific configurations and components.
  - **ui.content:** Stores site content, templates, and structure.
  - **ui.frontend:** Holds frontend assets like JavaScript and CSS.

## 8. Why we are using run mode?

- Run modes allow environment-specific configurations.
- Example: Development (dev), Staging (stage), and Production (prod).
- Helps in managing settings without modifying the codebase.

## 9. What is publish env?

- The **Publish Environment** in AEM serves content to end-users.
- It is optimized for delivering published content and does not allow direct authoring.

## 10. Why we are using dispatcher?

- The **Dispatcher** is used in AEM for caching and load balancing.
- It improves performance by serving cached content and enhances security by filtering requests.

## 11. From where can we access the CRX/DE?

- CRX/DE is the content repository interface in AEM.
- Accessible at: `http://<host>:<port>/crx/de`.