

MAVEN AND AEM CONCEPTS

1. Maven Lifecycle

Maven follows a lifecycle with predefined phases such as ``clean``, ``validate``, ``compile``, ``test``, ``package``, ``verify``, ``install``, and ``deploy``.

2. What is pom.xml and Why Do We Use It?

The ``pom.xml`` (Project Object Model) file is the core of a Maven project. It defines dependencies, plugins, goals, and project metadata. It helps in managing the build process and dependencies in a structured way.

3. How Dependencies Work?

Dependencies in Maven are libraries or modules required by the project. These are defined in ``pom.xml``, and Maven fetches them from repositories (like Maven Central).

4. Check the Maven Repository

Maven repositories are storage locations for dependencies. You can browse dependencies at <https://mvnrepository.com/>

5. How All Modules Build Using Maven?

Maven can build all modules in a multi-module project using the ``mvn install`` command at the parent module level. It builds each module based on dependency hierarchy.

6. Can We Build a Specific Module?

Yes, you can build a specific module by navigating to its folder and running ``mvn install``. Alternatively, use ``mvn install -pl module-name -am``.

7. Role of ui.apps, ui.content, and ui.frontend Folders

- `ui.apps`: Contains AEM-specific components, templates, and configurations.
- `ui.content`: Stores AEM content such as pages and sample content.
- `ui.frontend`: Manages the front-end (CSS, JavaScript, clientlibs).

8. Why We Use Run Mode?

Run modes in AEM help define environment-specific configurations (e.g., development, staging, production). They allow customization without modifying the core codebase.

9. What Is the Publish Environment?

The Publish environment in AEM serves final content to end users. It ensures that only approved content is accessible on the live website.

10. Why We Use Dispatcher?

The Dispatcher is a caching and security tool for AEM. It helps improve performance and protects against unauthorized access by caching content and filtering requests.

11. From Where Can We Access crx/de?

The `crx/de` (CRX Development Environment) can be accessed at:

- `http://localhost:4502/crx/de` (Author)
- `http://localhost:4503/crx/de` (Publish)