

AEM and MAVEN Q&A:

What is the purpose of the core module in AEM?

-The core module holds all the Java code like servlets. It keeps back-end logic separate from the front-end, and helps in reusing logic across multiple components.

What kind of files and code can be found in the core folder?

-Java classes for models, services, and servlets, plus utility classes and unit tests. It contains pom.xml to manage dependencies.

Explain the role of ui.apps in AEM projects.

-It contains all front-end stuff—components, templates, dialogs, and client libraries like CSS and JS. Stores content structure under /apps and helps in defining how components interact with users.

How are components structured in the ui.apps folder?

-Each component has an HTL file for UI, a .content.xml file for definitions, a dialog file for editing, and sometimes clientlibs.

Hello World Component:

- Where is the Hello World component located in both core and ui.apps?

- Core: com.project.core.models.
Ui.apps:/app/project/components/helloworld.

- Explain the Java class (in core) for the Hello World component.

-The HelloWorldModel.java class is an AEM Sling Model that retrieves properties and provides logic for the component.

- How does the HTL script work in ui.apps for Hello World?

-The HTL file **helloworld.html** renders the output using the Sling Model. `${model.message}` retrieves the value from the HelloWorldModel.java class. If no value is set, it displays "Hello, World!".

- How are properties and dialogs defined for this component?

-AEM components typically define properties using `cq:dialog` and `.content.xml`.

`cq:editConfig.xml` defines component behavior in the editor Can use `Sightly` use-api to link Java logic with HTL.

What are the different types of AEM modules (core, ui.apps, ui.content, etc.)?

- core - Java logic (models, services, servlets)
- ui.apps - Components, templates, dialogs, clientlibs
- ui.content - Sample content and content packages
- ui.config - OSGi configurations
- all -Packages everything together

How does Maven build these modules?

-Maven compiles, tests, and packages code before deploying it to AEM. Uses multi-module builds to handle different parts separately and Ensures all modules are built in the correct order.

Explain the build lifecycle of Maven in the context of AEM.

-It goes through steps like compile, test, package, install, and deploy to get everything ready for AEM. Runs unit tests automatically before packaging. Uses Phases & Goals to control builds.

How are dependencies managed in pom.xml?

-pom.xml lists all required dependencies, like AEM's Uber Jar and other OSGi bundles. Keeps track of third-party libraries.Uses dependency management to avoid version conflicts.

Why is Maven used instead of other build tools?

-It automates builds, manages dependencies easily, and integrates well with AEM. Follows a structured approach with XML-based configuration.Supports plugins to extend its functionality.

What advantages does Maven offer for AEM development?

- Standardized project setup
- Handles dependencies and OSGi bundles well
- Helps automate deployments
- Supports **continuous integration (CI/CD)** pipelines.
- Makes **collaborating** easier by defining dependencies centrally.

How does Maven help in managing dependencies and plugins in AEM projects?

-It pulls required libraries and plugins automatically from repositories. Reduces manual effort in downloading dependencies. Uses plugin goals to extend functionality.

What does mvn clean install do in an AEM project?

-Cleans the project, compiles the code, runs tests, and installs the package. Ensures a fresh build by removing old artifacts and Helps in local testing before deploying to AEM.

How to deploy packages directly to AEM using Maven commands?

-Use mvn -PautoInstallPackage clean install to deploy directly to AEM. Speeds up deployment by automating package installation. Can be used with AEM as a Cloud Service.

Explain the purpose of different Maven profiles in AEM (autoInstallPackage, autoInstallBundle).

- autoInstallPackage - Installs the full package
- autoInstallBundle - Installs just the Java bundle
- integration-test - Runs integration tests

What is the purpose of dumplib in AEM?

-Helps debug client libraries (JS, CSS) by showing their structure and dependencies. Shows **which** clientlibs are loaded on a page. Useful for **troubleshooting** missing styles or scripts.

How can you view client libraries using dumplib?

-Go to /libs/granite/ui/content/dumplib.html in your browser to see all clientlibs. Shows dependencies between clientlibs. Can be filtered to see specific categories.

Explain how client libraries are structured in AEM.

-A folder with css.txt, js.txt, and a resources directory for assets like images and fonts. Supports theme-based structuring for better organization. Uses allowProxy=true to make clientlibs accessible from /etc.clientlibs.