What is the purpose of the core module in AEM?

Ans: The core module contains all the backend or business logic and it handles the data processing.It contains the Java classes.

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

Ans: It includes Java classes, OSGi configurations,unit test files,etc..

Explain the role of ui.apps in AEM projects.

Ans: The ui.apps module contains all the front-end code required for the application like components, templates, and client libraries.This module defines the structure of the component.

How are components structured in the ui.apps folder?

Ans: ui.apps/src/main/content/jcr\_root/apps/project-name/components/

**Hello World Component:**

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

Ans: In core: core/src/main/java/com/project/models/HelloWorldModel.java

In ui.apps:

ui.apps/src/main/content/jcr\_root/apps/project/components/helloworld

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

Ans: The Java class in core for the Hello World component is a simple class, to support Sling Models. It contains methods to get data like a title or message. The data can come from component properties or other sources.

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

Ans: The HTL (HTML Template Language) script is used to display data from the Java class. It uses simple syntax like ${model.message} to show the message returned by the Java class.

* How are properties and dialogs defined for this component?

Ans: Properties are defined in the component's .content.xml file or through dialogs. Dialogs allow authors to input data like text, images, or links, and these values are stored as node properties in JCR (Java Content Repository).

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

Ans: AEM has different modules which are,

Core module: It contains the Java code.

Ui.apps module: It contains the component definitions (HTL, CSS, JS, dialogs).

Ui.content module: It contains content like pages, templates, and sample data.

Ui.tests module: It contains the test cases for components.

How does Maven build these modules?

Ans: Maven compiles code, packages it, and deploys it using defined steps. It follows a lifecycle: compile, test, package, install, and deploy. Each step runs plugins defined in pom.xml.

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

Ans: Maven follows phases like clean (delete old builds), compile (build Java code), package (create JARs), install (store locally), and deploy (upload to AEM). In AEM, it ensures proper component and bundle deployment.

How are dependencies managed in pom.xml?

Ans: pom.xml manages project dependencies (like AEM APIs, libraries) by fetching them from remote repositories. It ensures all required JARs are included during the build.

Why is Maven used instead of other build tools?

Ans: Maven is used because it’s standard in AEM. It manages dependencies, automates builds, and integrates well with Adobe’s tools. It also supports multi-module projects.

What advantages does Maven offer for AEM development?

Ans: Maven simplifies project setup, ensures consistent builds, manages libraries, and supports profiles for deploying code easily.

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

Ans: Maven’s pom.xml lists dependencies (libraries) and plugins. It fetches them automatically and keeps versions aligned.

What does mvn clean install do in an AEM project?

Ans: mvn clean install removes old build files (clean), compiles the code, packages it, and installs the output to the local repository.

How to deploy packages directly to AEM using Maven commands?

Ans: Use command mvn clean install -PautoInstallPackage -Dmaven.test.skip=true to build and deploy directly to AEM.

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

Ans: Profiles like autoInstallPackage deploy full content packages, while autoInstallBundle deploys only the Java code. Profiles helps to control what gets deployed.

What is the purpose of dumplibs in AEM?

Ans: Dumplibs in AEM lists client libraries (JS and CSS) loaded on a page. It helps check if the right libraries are included.

How can you view client libraries using dumplibs?

Ans: We can use /libs/granite/ui/content/dumplibs.html to view all loaded client libraries.

Explain how client libraries are structured in AEM.

Ans: ui.apps/src/main/content/jcr\_root/apps/project-name/clientlibs