AEM Questions

What is the purpose of the core module in AEM?

It contains backend logic, including Java classes implementing business logic and Sling models to interact with AEM components.

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

The core folder contains Java classes, Sling models, OSGi services, and servlets.

Explain the role of ui apps in AEM projects.

The ui apps module contains frontend resources like components, templates, dialogs, and client libraries. It defines the application structure .

How are components structured in the ui apps folder?

Components are structured under /apps/myTraining/components/. Each component has an HTML template (HTL), a JavaScript file, a dialog.xml file for editing properties, and a .content.xml for node definitions.

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

In the core module: /core/src/main/java/com.myTraining.core/models/HelloWorldModel.java
In the ui.apps module: /apps/myTraining/components/helloworld/

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

The Java class (Sling Model) retrieves and processes data for the component. It uses annotations like @Model and @postConstruct.It has @slingObject which is defined with message later it will return Hello World resource type and the current page

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

HTL (HTML Template Language) retrieves and displays values from the Java model using expression language syntax .it is having Hello world component ,Text property ,Model message

How are properties and dialogs defined for this component?

Dialogs are defined in dialog.xml using Granite UI components. Properties are stored in JCR nodes and accessed via Sling models.

What are the different types of AEM modules?

- core: Contains Java code and business logic
- ui.apps: Holds components, templates, and client libraries
- ui.content: Stores content structures like pages and configurations
- it.tests: Integration tests
- all: Aggregates all modules into a deployable package

How does Maven build these modules?

Maven uses the POM hierarchy to compile Java code, package components, and generate AEM packages for deployment.

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

Maven follows phases like validate, compile, test, package, install, and deploy. For AEM, it compiles Java code, assembles content packages, and deploys them.

How are dependencies managed in pom.xml?

Dependencies are managed in <dependencies> using groupId, artifactId, and version. AEM projects also use <dependencyManagement> to control versions.

Why is Maven used instead of other build tools?

Maven automates dependency management, enforces a standardized project structure, and integrates well with AEM's package management system.

What advantages does Maven offer for AEM development?

- Centralized dependency management
- Automated builds and deployments
- Support for multi-module projects
- Integration with AEM's content packaging system

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

Maven fetches dependencies from repositories and configures AEM-specific plugins like content-package-maven-plugin for building and deploying packages.

What does mvn clean install do in an AEM project?

It cleans old files and makes new files for AEM.

How to deploy packages directly to AEM using Maven commands?

Use mvn clean install -PautoInstallPackage to deploy the package to AEM.

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

- autoInstallPackage: Deploys the full content package to AEM
- autoInstallBundle: Deploys only the OSGi bundle (Java code) without affecting content

What is the purpose of dumplibs in AEM?

dumplibs is used to debug and analyze client libraries. It helps in identifying dependencies, categories, and paths of client libraries

How can you view client libraries using dumplibs?

Access http://localhost:4502/libs/granite/ui/content/dumplibs.html to see the structure and dependencies of client libraries.

Explain how client libraries are structured in AEM.

Client libraries are stored under /apps/project/clientlibs/. They are categorized using categories, dependencies, and embed properties to manage CSS, JS, and assets efficiently.