DAY – 2 TASKS

What is the purpose of the core module in AEM?

The core module in AEM contains backend logic, including Java classes for components and services. It handles business logic and interacts with AEM APIs.

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

The core folder contains Java classes, Sling models, OSGi services, and utility classes. These files define the backend functionality of AEM components.

Explain the role of ui.apps in AEM projects.

The ui.apps module contains front-end code like HTL scripts, CSS, JavaScript, and component definitions. It defines how components are rendered and configured in AEM.

How are components structured in the ui.apps folder?

Components in ui.apps are structured in /apps/project/components, each having an html file (HTL script), dialog.xml for configurations, and cq:dialog for authoring UI.

**Hello World Component:**

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

In core, it's a Java class in com.project.core.models. In ui.apps, it's under /apps/project/components/helloworld.

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

**Java Class (Core)** defines a Sling model with annotated properties, retrieving data from AEM resources.

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

HTL Script (ui.apps): Uses ${model.property} to bind backend data dynamically to the frontend.

1. How are properties and dialogs defined for this component?

Properties & Dialogs: Defined in cq:dialog under /apps/project/components/helloworld, allowing author input.

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

Modules include

* core (Java logic)
* ui.apps (frontend and components)
* ui.content (default content)
* ui.config (configurations)

How does Maven build these modules?

Maven compiles Java (core), processes frontend files (ui.apps), and packages them as an AEM installable package (all).

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

## Maven Build Lifecycle in AEM

* Validate: Checks project structure and configuration before the build.
* Compile: Compiles the Java source code into bytecode.
* Package: Bundles compiled code into a JAR or AEM content package.
* Install: Installs the package into the local Maven repository.
* Deploy: Deploys the package to a remote repository or AEM instance.

How are dependencies managed in pom.xml?

Dependencies are managed using <dependencies> and <dependencyManagement>, ensuring version control and resolving library conflicts.

Why is Maven used instead of other build tools?

Maven automates builds, manages dependencies efficiently, and integrates seamlessly with AEM’s OSGi framework.

What advantages does Maven offer for AEM development?

It simplifies dependency management, enforces project structure, and automates package deployment to AEM.

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

Maven centralizes dependency versions and uses plugins (e.g., content-package-maven-plugin) for AEM-specific tasks.

What does mvn clean install do in an AEM project?

It cleans, compiles, tests, and packages the project, generating deployable content and OSGi bundles.

How to deploy packages directly to AEM using Maven commands?

Use mvn -PautoInstallPackage install to deploy code directly to AEM.

CMD : mvn clean install -PautoInstallPackage -Dmaven.test.skip=true

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

* autoInstallPackage: Deploys full packages.
* autoInstallBundle: Deploys OSGi bundles only.

What is the purpose of dumplibs in AEM?

dumplibs helps debug and list all client libraries (/libs/granite/ui/content/dumplibs).

How can you view client libraries using dumplibs?

Access /libs/granite/ui/content/dumplibs.html to see registered client libraries and their dependencies.

Explain how client libraries are structured in AEM.

Client libraries in AEM are stored under /apps/project/clientlibs. They contain JavaScript, CSS, and other assets for frontend styling and functionality. The categories property defines how and when these libraries are loaded. They help manage dependencies and optimize performance.