**1.What is the purpose of the core module in AEM?**

The core module in AEM contains Java-based components, OSGi services, and utility classes that provide the foundation for building AEM applications.

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

* Java classes(.java files)
* Servlets
* Filters
* Utility classes
* Configuration files like XML, Properties Files, etc…..

**3.Explain the role of ui.apps in AEM projects.**

**ui.apps** defines the UI structure and layout of the AEM application.

**4.How are components structured in the ui.apps folder?**

They are structured with each component having its own folder containing subfolders and files.

**5.Hello World Component:**

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

In **core**, the Hello World component is located in a Java package, such as com.example.aem.core.components.helloworld.

In **ui.apps**, the Hello World component is located in a folder structure, such as ui.apps/src/main/content/jcr\_root/apps/myapp/components/helloworld.

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

Java class extends WCM or implements an interface, returns a "Hello World" message. By extending WCM class, developers can create custom AEM components that are easy to use, maintain, and extend.

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

HTL script displays the message using data-sly-use and data-sly-text attributes.

* **How are properties and dialogs defined for this component?**

Properties defined in cq:Component node, dialogs defined in \_cq\_dialog node in CRXDE.

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

* **core module** contains Java classes, OSGi services, and utilities.
* **ui.apps module** contains HTL components, client-side JavaScript, and CSS.
* **ui.content module** contains sample content, such as pages, components, and assets.
* **it.tests module** contains integration tests for the project.
* **all module** is the parent module that aggregates all other modules.

**7.How does Maven build these modules?**

* Compiles Java classes in the core module.
* Packages compiled classes into JAR files.
* Installs packaged JAR files in the local Maven repository.
* Deploys packaged JAR files to AEM.

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

**Maven's build lifecycle consists around 8 phases:**

* validate: This Phase Validates project structure.
* compile: This Phase Compiles Java classes.
* test: This Phase Performs unit testing.
* package: Packages compiled classes into JAR files.
* integration-test: Performs integration testing.
* verify: Verifies project quality.
* install: Installs packaged JAR files in the local Maven repository.
* deploy: Deploys packaged JAR files to AEM.

**9.How are dependencies managed in pom.xml?**

Dependencies are managed using **groupId** Specifies the dependency's group ID, **artifactId** Specifies the dependency's artifact ID, **versionId** Specifies the dependency's version.

**10.Why is Maven used instead of other build tools?**

Maven is used because it simplifies project structure, and it also manages dependencies.

**11.What advantages does Maven offer for AEM development?**

* Simplifies project creation.
* Automatically manages dependencies.
* Ensures consistency across projects.

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

Maven manages by :

* pom.xml which defines project dependencies and plugins.
* Automatically downloads and manages dependencies.
* Configures and executes plugins.

**13.What does mvn clean install do in an AEM project?**

* It deletes the target directory.
* It Compiles Java classes.
* It Packages compiled classes into JAR files.
* Finally, it Installs packaged JAR files in the local Maven repository.

**14.How to deploy packages directly to AEM using Maven commands?**

To deploy packages to AEM using Maven:

* Configure the pom.xml file
* Run the **mvn clean install** command to build and package the project.
* Use the **mvn deploy** command to deploy the package to AEM.

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

* The purpose of **autoInstallPackage** is to automatically installs packages in AEM.
* The purpose of **autoInstallBundle** is to automatically installs bundles in AEM.

**16.What is the purpose of dumplibs in AEM?**

It’s main purpose is to help developers by allowing them to inspect client libraries like JavaScript and CSS files, debug client-side issues, and optimize client library performance

**17.How can you view client libraries using dumplibs?**

To view client libraries:

* Add **?debugClientLibs=true** to the AEM URL.
* Access the AEM page using the client library.
* View client library files via the dumplibs URL.

**18.Explain how client libraries are structured in AEM.**

Client libraries are structured by:

* Category
* Embedding
* Dependencies
* Compression
* Storage in clientlibs.