

### What is the purpose of the core module in AEM?

A. core module handles complex processing like perform backend operations.

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

A. java code, POM.xml

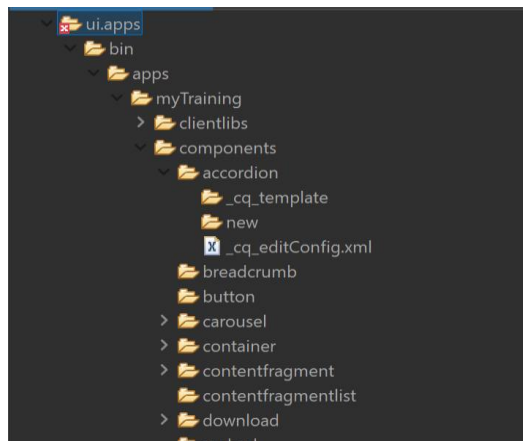
### Explain the role of ui.apps in AEM projects.

A. ui.apps is responsible for deploying front end and back end content into the aem repository.

It stores component and template definitions. Contains content.xml which defines how content is structured

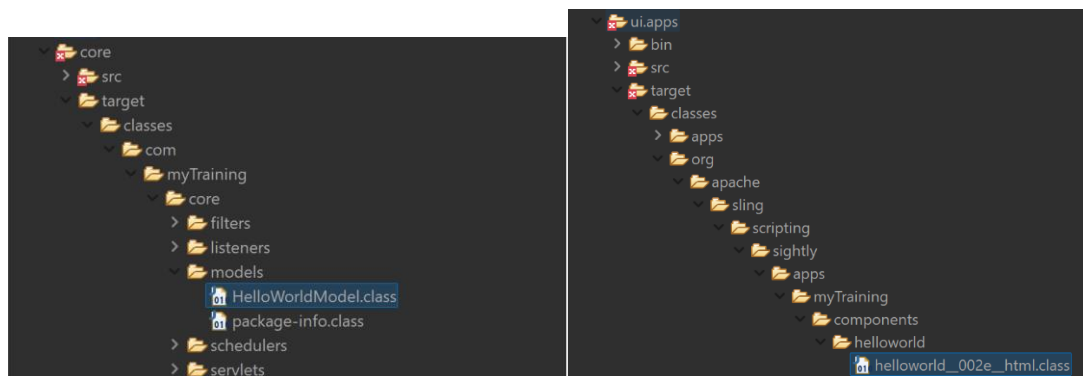
### How are components structured in the ui.apps folder?

A.



### Hello World Component:

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



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

It adapts to a resource

It injects message property from jcr

If no value is provided it returns default message

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

`${model.message}` calls the getter method in the sling model

If the message property is defined it displays that value otherwise displays default value

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

They are defined using touch ui

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

- A. all, core, dispatcher, it.tests, ui.apps, ui.apps. structure, ui.config, ui.content, ui.frontend, ui.tests

**How does Maven build these modules?**

- A. maven compiles java code, process ui apps and content modules, builds a deployable package, deployment to aem instance

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

- A.
- mvn clean
  - mvn validate
  - mvn compile
  - mvn test
  - mvn package
  - mvn verify
  - mvn install
  - mvn deploy

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

- A. dependencies are specified in the `<dependencies>` section

Provided – means that the dependency is available in the aem run time but not included in the final build

Maven repositories are used to resolve dependencies

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

- A.
- Standardized project structure
  - Dependency management
  - Integration with AEM

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

- A.
- consistent project structure
  - Automated build and deployment
  - Version dependency management

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

**A.** maven define all dependencies and plugins in a single pom.xml. this file is used to manage the project build and documentation

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

**A.** Clean the target directory, removing any previous builds and compile the project code.

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

**A.** mvn clean install -PautoInstallPackagePublish

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

**A.** autoInstallPackage: This profile is used to build and install the entire AEM package, including all content and code, to the local AEM instance.

autoInstallBundle: This profile is used to build and install only the OSGi bundle (Java code) to the local AEM instance

**What is the purpose of dumplib in AEM?**

**A.** The dumplib tool in AEM is used to view and manage client libraries. It provides a user-friendly interface to filter and inspect client libraries based on various criteria such as path, categories, channels, and type.

**How can you view client libraries using dumplib?**

**A.** <http://localhost:4502/libs/granite/ui/content/dumplib.html>

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

**A.** client libraries are used to manage and deploy CSS, JavaScript, and other client-side resources.

A client library folder is a repository node of type cq:ClientLibraryFolder. These folders can be placed within the /apps, /libs, and /etc subtrees of the repository.