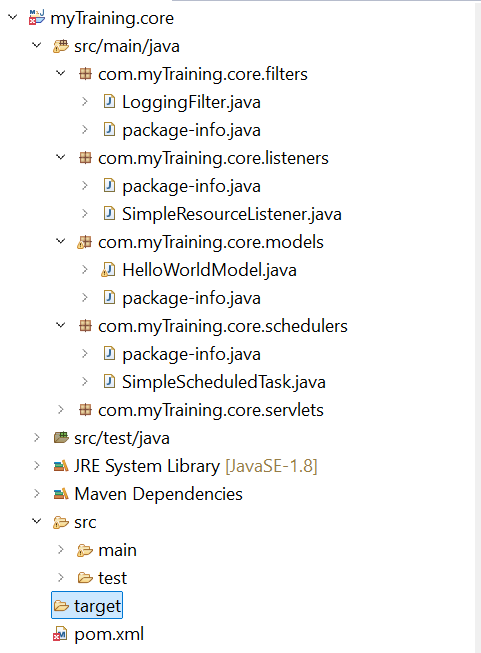
What is the purpose of the core module in AEM?

**Ans:** The CoreModule in AEM provides pre-built components and files like HelloWorldModel, LoggingFilter, SimpleResourceListener, SimpleScheduledTask, Maven Dependencies ,pom.xml,etc. It helps developers quickly build websites, ensures consistent design, and improves performance. It also reduces development time by resusing the components.

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

**Ans:** The Core Folder contains some java files such as HelloWorldModel, LoggingFilter, SimpleResourceListener, SimpleScheduledTask, Maven Dependencies and also pom.xml, src file which contains main file and test file, target file.

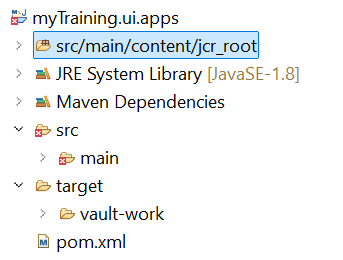


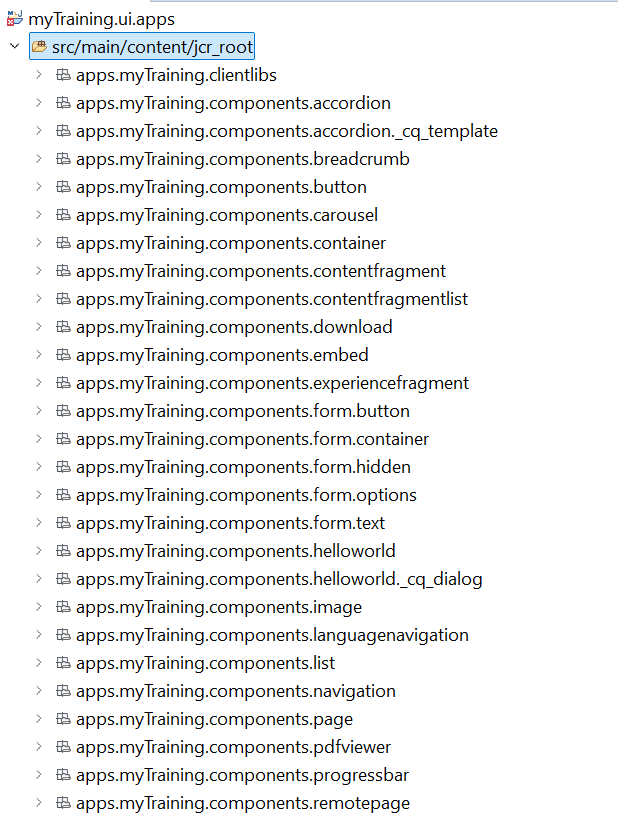
Explain the role of ui.apps in AEM projects.

**Ans:** In AEM projects, the ui.apps module is responsible for deploying content and configurations to the AEM instance. It contains all the necessary files required for the site structure,components, templates, and client-side assets. It contains clientlibs, errorHandler and other components. It also contains src and target file, Maven dependencies, pom.xml.

How are components structured in the ui.apps folder?

**Ans:** Ui.apps contains clientlibs, errorhandler and other components inside jcr\_root which is in src/main/content. It also contains src and target file, Maven dependencies, pom.xml.





**Hello World Component:**

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

**Ans:** In both core and ui.apps the Hello World component is located in

src/main/myTraining

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

**Ans:** The HelloWorldModel.java is used to return the message. The message contains a small message like Hello World, Resource type, Current type. So by using getMessage() method, the message is passed.

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

**Ans:**  HTL Automatically Binds to the Sling Model and fetches title and message from it and automatically connects to HelloWorldModel.java in the core module because of the component binding in AEM.

* How are properties and dialogs defined for this component?

**Ans:** In AEM, properties store component data in the JCR, and dialogs allow authors to enter content. Dialogs are defined in \_cq\_dialog/.content.xml using XML. The stored values can be accessed in HTL (${properties.title}) or through Java Sling Models (@ValueMapValue private String title)

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

**Ans:** Core Module, ui.apps Module, ui.content Module, ui.config Module, ui.tests Module and All Module.

How does Maven build these modules?

**Ans:** Maven builds AEM projects by compiling Java (core), packaging components (ui.apps), and bundling content (ui.content). The all module combines everything into a deployable package.

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

**Ans:** Maven follows a build lifecycle to compile, test, package, and deploy an AEM project step by step. The three main lifecycles are:

Clean : Deletes previous builds  
 Default : Builds the project  
 Site : Generates documentation

How are dependencies managed in pom.xml?

**Ans:** Maven manages dependencies in pom.xml by defining them in the <dependencies> section. AEM projects often use a parent POM to centralize dependencies. Different scopes like provided, compile, and test control how dependencies are included.

Why is Maven used instead of other build tools?

**Ans:** Maven is the preferred build tool for AEM development because it simplifies dependency management, project structure, and automated builds.

What advantages does Maven offer for AEM development?

**Ans:**  Maven is ideal for AEM development because it provides a standardized project structure, automates dependency management, and simplifies the build lifecycle. It also integrates directly with AEM using Adobe’s official plugins.

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

**Ans:** Maven simplifies dependency and plugin management in AEM projects by automatically handling library versions, fetching required packages, and integrating with Adobe’s tools.

What does mvn clean install do in an AEM project?

**Ans:** The command mvn clean install is used in AEM projects to build the entire project, clean old files, and install the new build into the local Maven repository.

How to deploy packages directly to AEM using Maven commands?

**Ans:** To deploy packages directly to AEM using Maven, we use mvn clean install -P autoInstallPackage to install the full package or -PautoInstallBundle for only backend code

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

**Ans:** autoInstallPackage: Deploys entire AEM package (components, templates, dialogs, clientlibs)

autoInstallBundle: Deploys only the OSGi bundle (Java code in core module)

What is the purpose of dumplibs in AEM?

**Ans:** Dumplibs is a built-in debugging tool that helps developers analyze and troubleshoot Client Libraries (clientlibs).

How can you view client libraries using dumplibs?

**Ans:** To view client libraries in AEM, visit http://localhost:4502/libs/granite/ui/content/dumplibs.html. This page lists all available clientlibs, their dependencies, and their status.

Explain how client libraries are structured in AEM.

**Ans:** In AEM, client libraries (clientlibs) are used to manage CSS, JavaScript, and assets. They are structured in folders inside ui.apps, with categories defined in .content.xml. We include them in components using HTL or JSP.