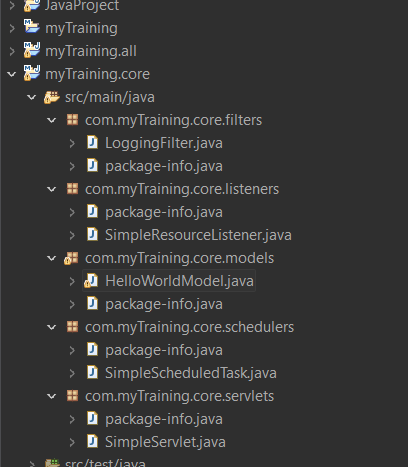
**4/3/25**

**Day 2**

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

Core module in AEM contains the backend logic for the project. It has java classes required for the backend logic. To fetch and process data it has sling models, OSGi services, servlets.

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

****

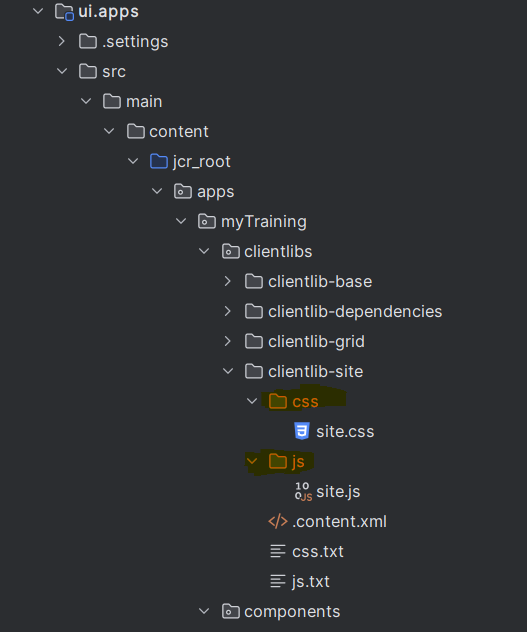
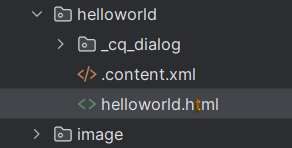
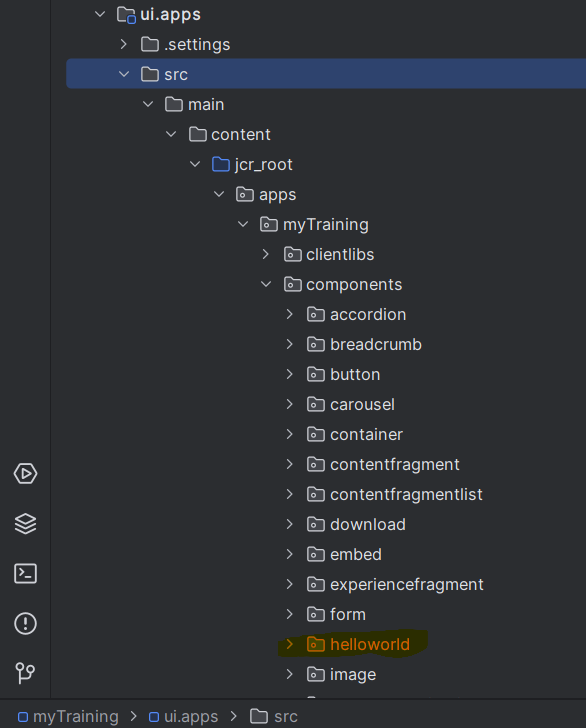
* Sling Models -to fetch data from AEM
* OSGi Services -handles business logic
* Servlets-process request
* Java classes-backend logics

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

The ui.apps module contains the frontend code (HTL, JavaScript, CSS) and component definitions used in the AEM project. It defines how components look and function

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

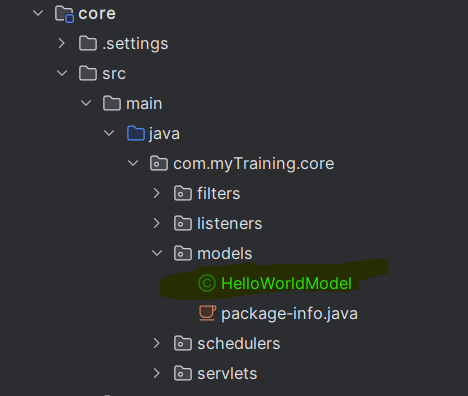
In ui.apps folder, components are stored inside jcr\_root/apps/myTraining/components, with each component having its own folder. Each component contains HTL for display, \_cq\_dialog for editing, and .content.xml for configuration. This setup makes components reusable and easy to manage in AEM



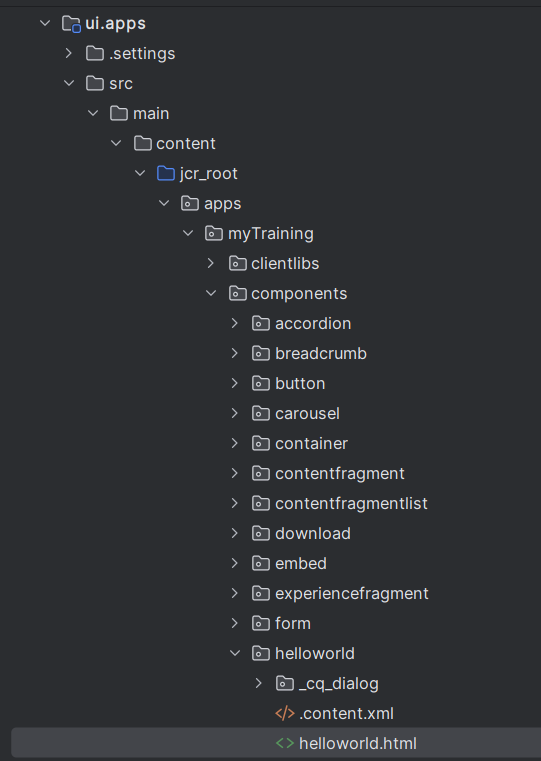
**5)Hello World Component:**

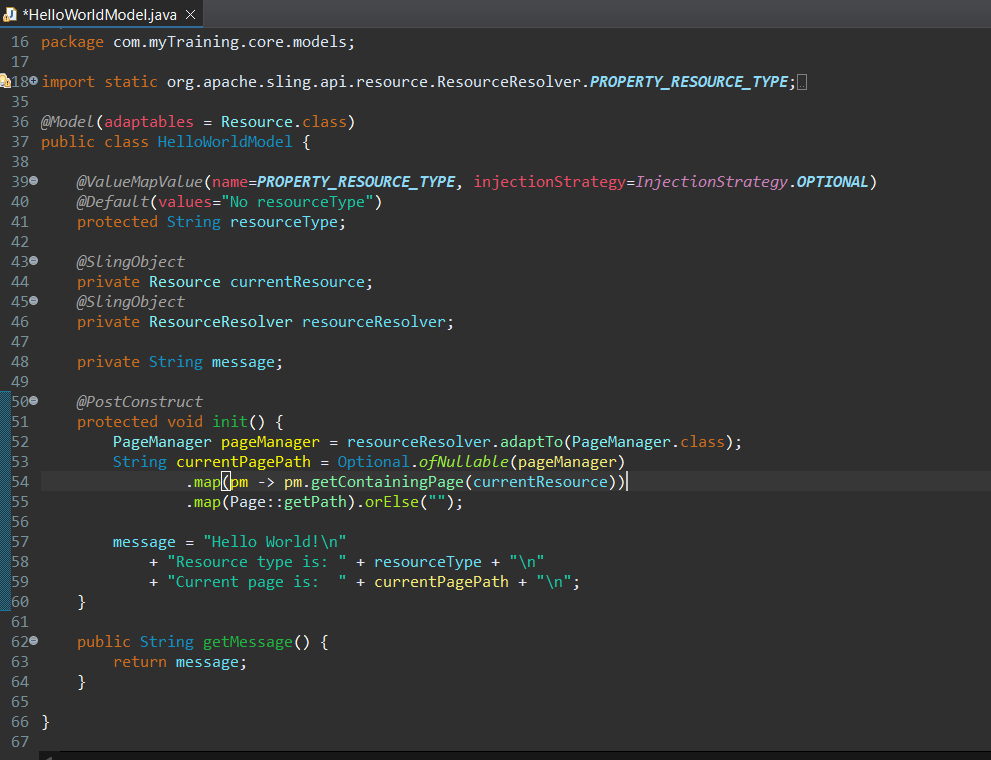
* **Where is the Hello World component located in both core and ui.apps?**

**Core Module** (/core/src/main/java/com.myTraining.core/models/HelloWorldModel.java)

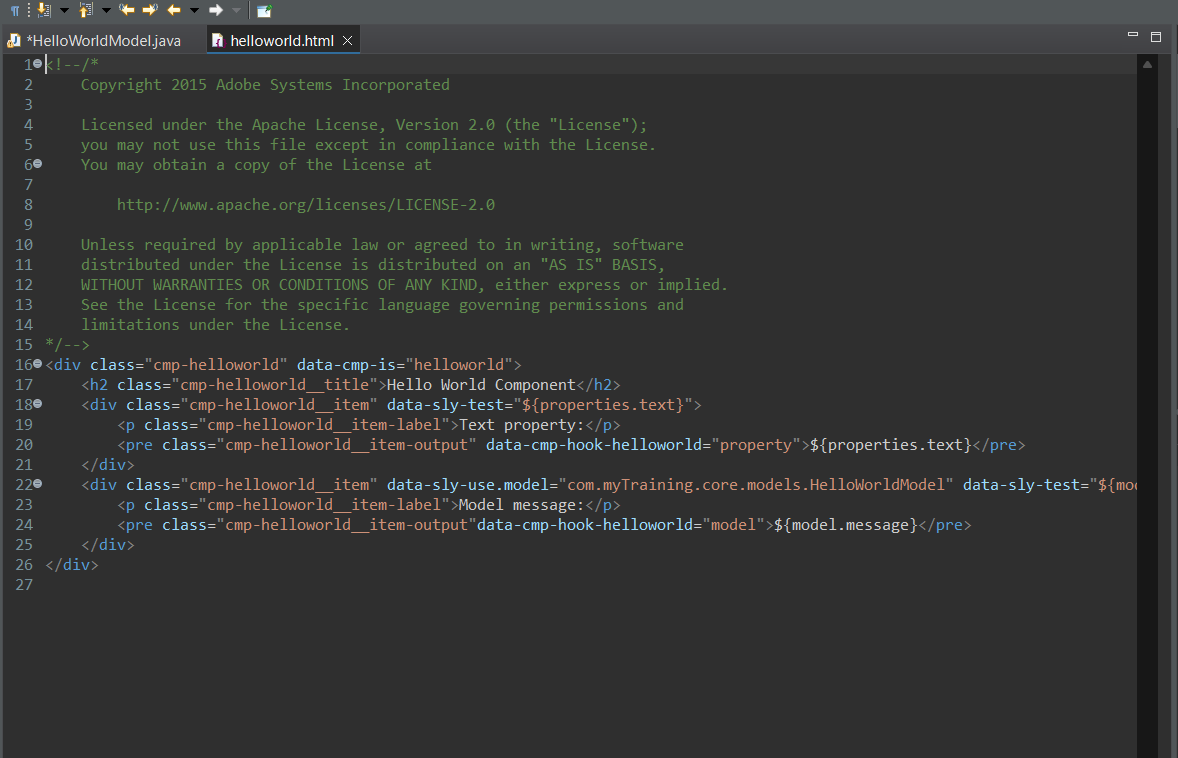


**UI Apps Module** (ui.apps/src/main/content/jcr\_root/apps/myTraining/components/helloWorld/)



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

The HelloWorldModel.java in the core module is a sling model that fetches data and makes it available to the frontend

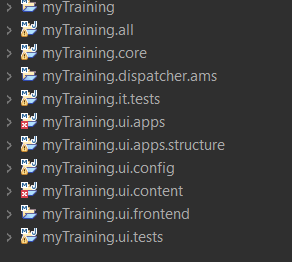
* **How does the HTL script work in ui.apps for Hello World?**
* ****

The HTL script in ui.apps fetches data from the HelloWorldModel and displays it on the page.

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

In \_cq\_dialog.xml, the dialog includes a text field that allows authors to enter a custom message. This input is stored in the JCR and used by the component.

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

****

- core – Contains backend logic (sling Models, servlets, OSGi services).

- ui.apps – Stores frontend components, dialogs, and client libraries.

- ui.content – Manages actual website content and structure.

- dispatcher – Handles caching and request forwarding for performance.

- all – Bundles everything into a single installable package.

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

Maven compiles the java code, packages UI components, and creates a deployable ZIP for AEM. Each module is built separately and then combined

Commands:

* mvn clean install
* mvn clean install -PautoInstallSinglePackage
* mvn clean install -PautoInstallPackage.
* mvn clean install -PautoInstallAll.

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

* compile – Converts Java code into .class files.
* test – Runs unit tests.
* package – Creates jar and zip
* deploy – Uploads the package to AEM

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

Dependencies are added in pom.xml using <dependencies> tags. Maven automatically downloads and manages them.

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

Maven simplifies dependency management, automates builds, and integrates well with AEM.

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

* Manages the dependencies automatically
* Build automation and deployment
* Organizes projects into modules

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

It automatically downloads and maintains the required libraries and APIs through its central repository. It uses the pom.xml file to define dependencies, ensuring consistency across different environments

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

**mvn clean install command** Compiles Java code, Packages frontend components, Deploys the package to AEM

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

To deploy packages directly to AEM use

-**mvn clean install -PautoInstallPackage------**It automatically installs the project inside AEM

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

mvn clean install -PautoInstallPackage ------Installs everything (backend & frontend).

mvn clean install -PautoInstallBundle ------ Installs only OSGi bundle (backend).

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

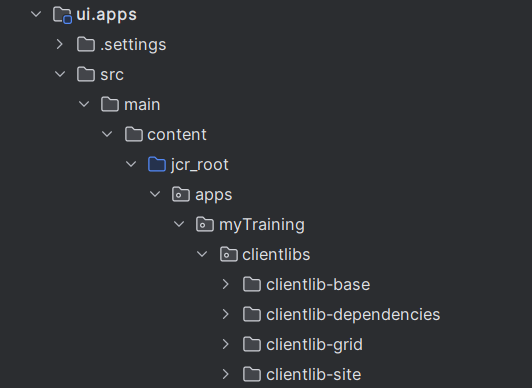
dumplibs is a tool used to debug and analyze client libraries (clientlibs). It checks how CSS and JavaScript files are loading in AEM. It helps to find issues if styles or scripts are not working properly

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

Open your browser and navigate to **http://localhost:4502/libs/granite/ui/content/dumplibs.html**

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

Inside ui.apps module, src/main/content/jcr\_root directory contains the apps/myTraining folder. Within this, the clientlibs folder has multiple client libraries, like clientlib-base, clientlib-dependencies, clientlib-grid, and clientlib-site. Each subfolders are used to manage different styles, scripts, and dependencies for the project.

****