TASK-02

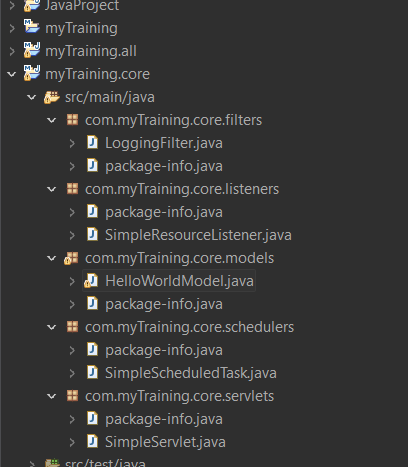
1.What is the purpose of the core module in AEM?

Core Module is responsible for backend business logic in AEM.

* Sling: It fetches the JCR content for the component.
* OSGI: It decouples business logic from components.
* Custom Servlets: It provides API endpoints.
* Schedulers: Runs tasks at fixed intervals.
* Event Listener: Automate tasks based on JCR changes.

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

Core folder contains models, filters, schedulers, event listeners, servlets and pom.xml file.



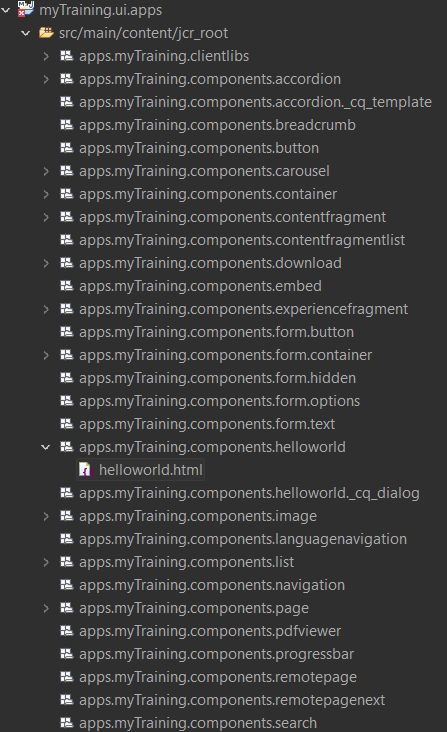
3.Explain the role of ui.apps in AEM projects.

The ui.apps module in an AEM project contains everything needed for the frontend.

* It stores AEM components.
* It contains page templates.
* Manage client libraries and include content configurations.

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

It is structured in hierarchical way under the apps folder.



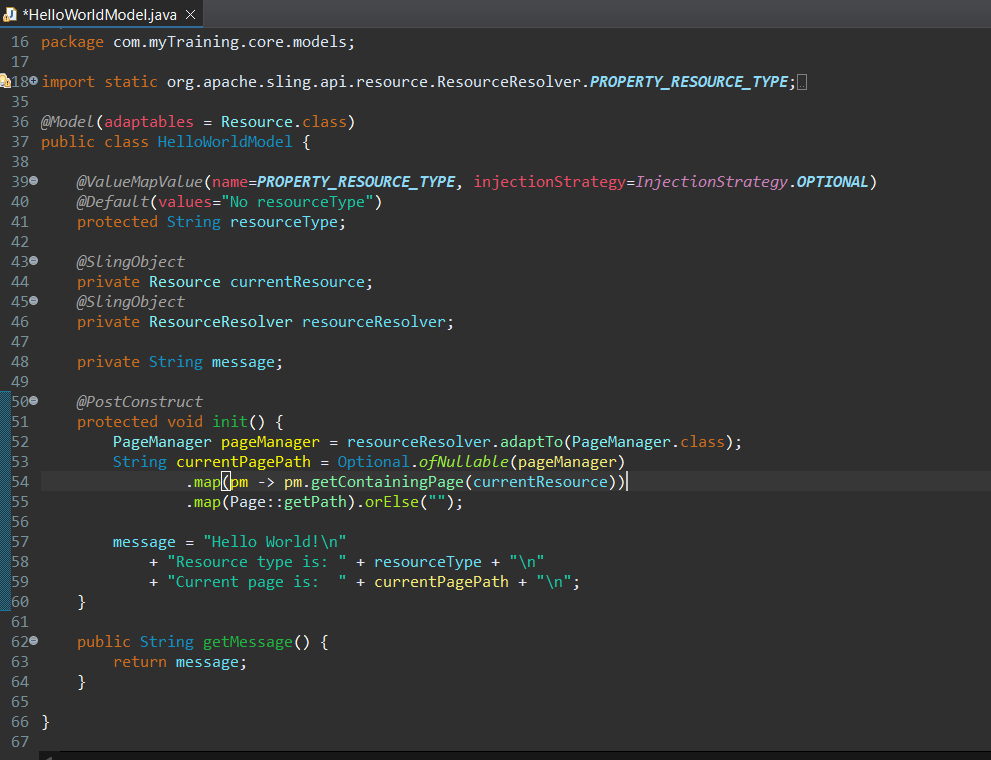
**5.Hello World Component:**

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

* In core, Hello World Component is present in models folder which is responsible for backend logic.
* In ui.apps, Hello World Component is present in components folder which is responsible for frontend logic.

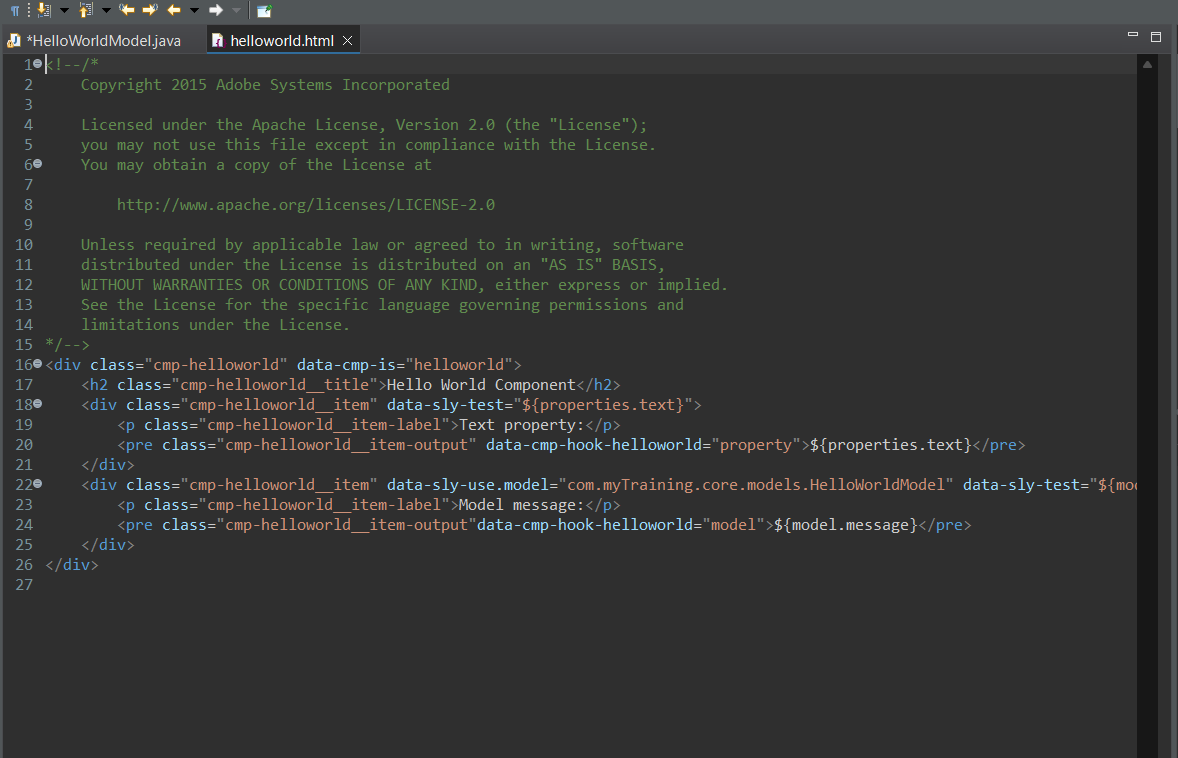
1. Explain the Java class (in core) for the Hello World component.

* Java class which is present in core module is a Sling model that fetches data from the Java Content Repository.



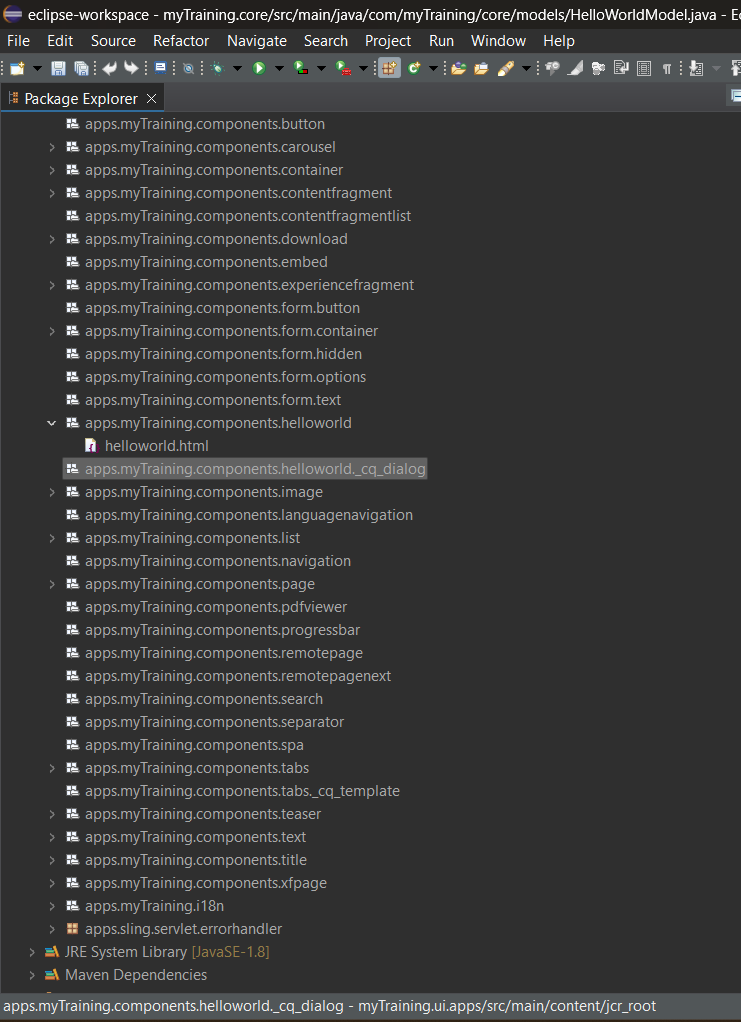
1. How does the HTL script work in ui.apps for Hello World?

* The HTL script in ui.apps is responsible for rendering the Hello World components frontend UI using data from the Sling model.
* It simplifies frontend development by cleanly separating UI and backend logic.



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

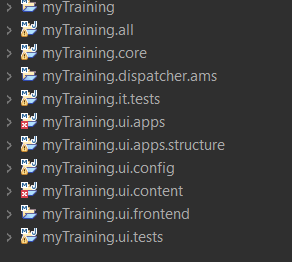
* In AEM, properties store content entered by authors, and dialogs provide a UI in the editor for managing these properties.
* Dialog: It allows content authors to enter data.
* Properties: It saves the content in the JCR repository.



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

It contains core, ui.apps, disptacher.arms, it.tests, ui.apps, apps.structure, ui.config, ui.content, ui.tests, ui.frontend .

* Core Module: Backend logic.
* ui.apps Module: Frontend and Component definitions.
* ui.content Module: It contains content for deployment.
* it.tests Module: It is used for integration tests.
* all Module: It is used for build and deployment.



7.How does Maven build these modules?

* Maven uses a multi-module project structure in AEM, where each module has a specific role. But the parent POM file defines dependencies, version, plugins etc.
* It reads the parent Pom to find all submodules and builds core and process ui.apps and run tests and bundle everything in all for deployment.

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.

It contains different phases such as

* clean - Removes old build files.
* validate - Checks if the project structure is correct.
* compile - Compiles Java classes
* test - Runs unit tests
* package - Bundles everything into an AEM .zip package.
* install - Installs the package to the local repository.
* deploy - Deploys the package to AEM.

9.How are dependencies managed in pom.xml?

* In Maven, dependencies are managed using the <dependencies> section which is inside the pom.xml file.
* AEM projects use dependencies to include libraries like sling, OSGI for backend development.

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

* Maven is used because it automates dependency management, ensuring all required libraries are downloaded and versioned correctly.
* It follows a structured build lifecycle, making it easy to compile, test, and package AEM projects.
* It supports modular project structures, which is crucial for large-scale AEM applications.

11.What advantages does Maven offer for AEM development?

* Maven simplifies AEM development by handling dependency management. so we don’t need to manually download JARs.
* It provides a consistent build process, making it easy to compile, test, and deploy AEM projects.
* It supports profiles for different environments, so we can deploy to local, dev, or prod seamlessly.

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

* Maven automatically downloads and manages dependencies, ensuring all AEM modules use the correct versions without conflicts.
* It also integrates plugins to build, deploy, and install AEM packages smoothly.

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

* It clears old build files, compiles Java code, packages the AEM project, and installs it locally.
* It ensures a fresh build and prepares the project for deployment.

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

* We can deploy AEM packages using “mvn clean install -PautoInstallSinglePackage” which builds and installs the full package to AEM.
* If we only want to deploy content and components, we can use “mvn clean install -PautoInstallPackage”.
* For deploying all modules separately, “mvn clean install -PautoInstallAll” does the job.

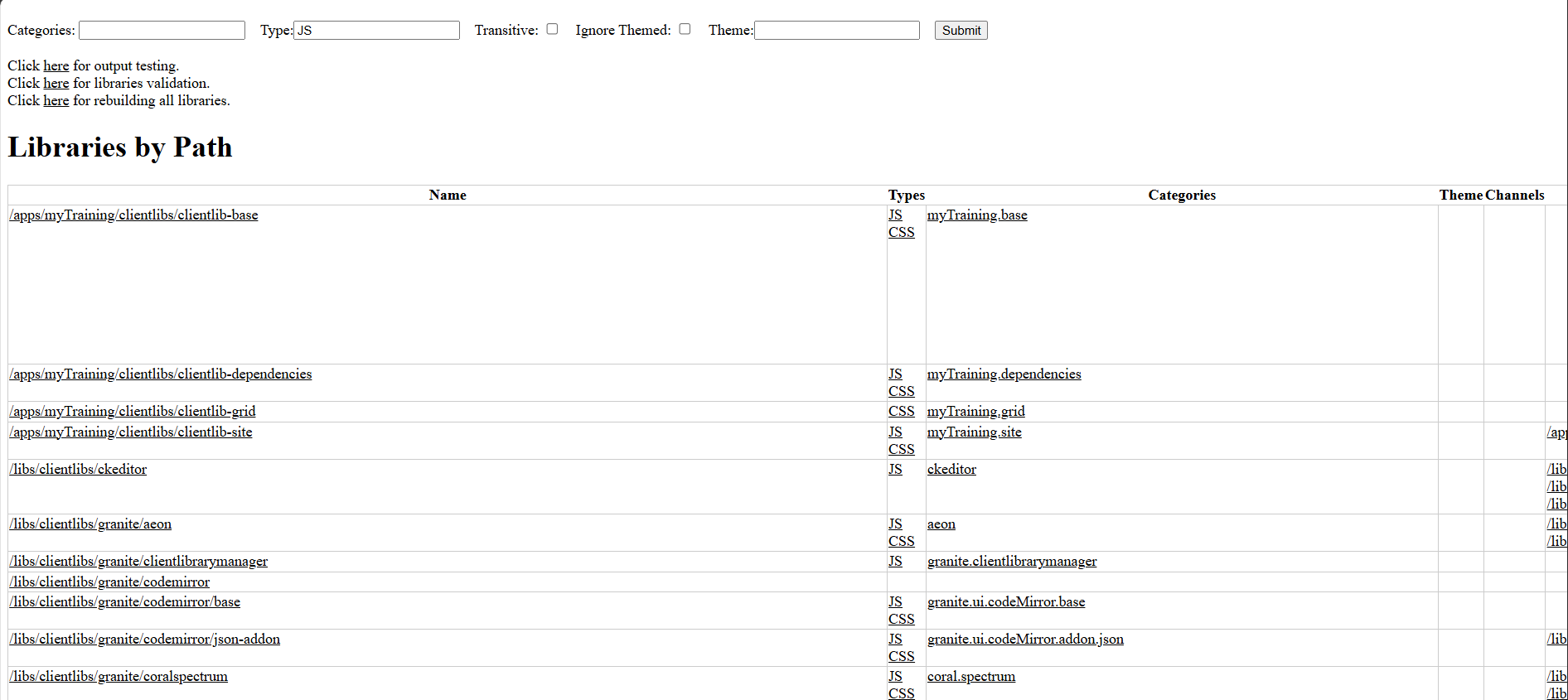
15.Explain the purpose of different Maven profiles in AEM (autoInstallPackage, autoInstallBundle).

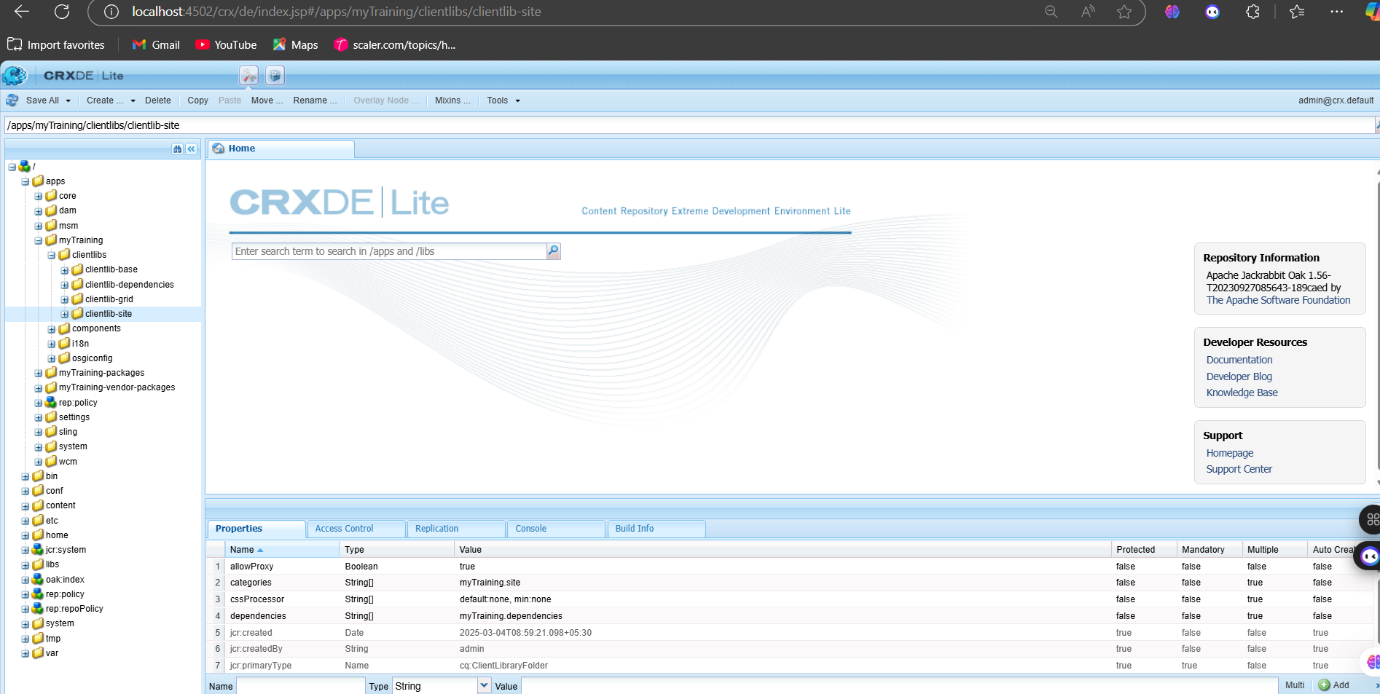
* Maven profiles in AEM help with controlled deployments based on the need.
* autoInstallPackage deploys only the content and component packages.
* autoInstallBundle installs the OSGi bundle.

This makes it easier to deploy specific parts without rebuilding everything.

16.What is the purpose of dumplibs in AEM?

* In AEM, dumplibs is used to check which clientlibs are loading, their dependencies, and for any issues. It helpful for debugging clientlibs not loading properly.





17.How can you view client libraries using dumplibs?

* We can view client libraries by using “/lis/grantie/ui/content/dumplibs.html” in our AEM instance.
* It shows all loaded clientlibs, their dependencies, and helps debug missing or incorrect client-side assets.

18.Explain how client libraries are structured in AEM.

* In AEM, client libraries are stored under “/apps/project-name/clientlibs” and grouped into categories like CSS, JS, and dependencies.
* Each clientlib has a cq:ClientLibraryFolder node with a categories property to define how it’s loaded.