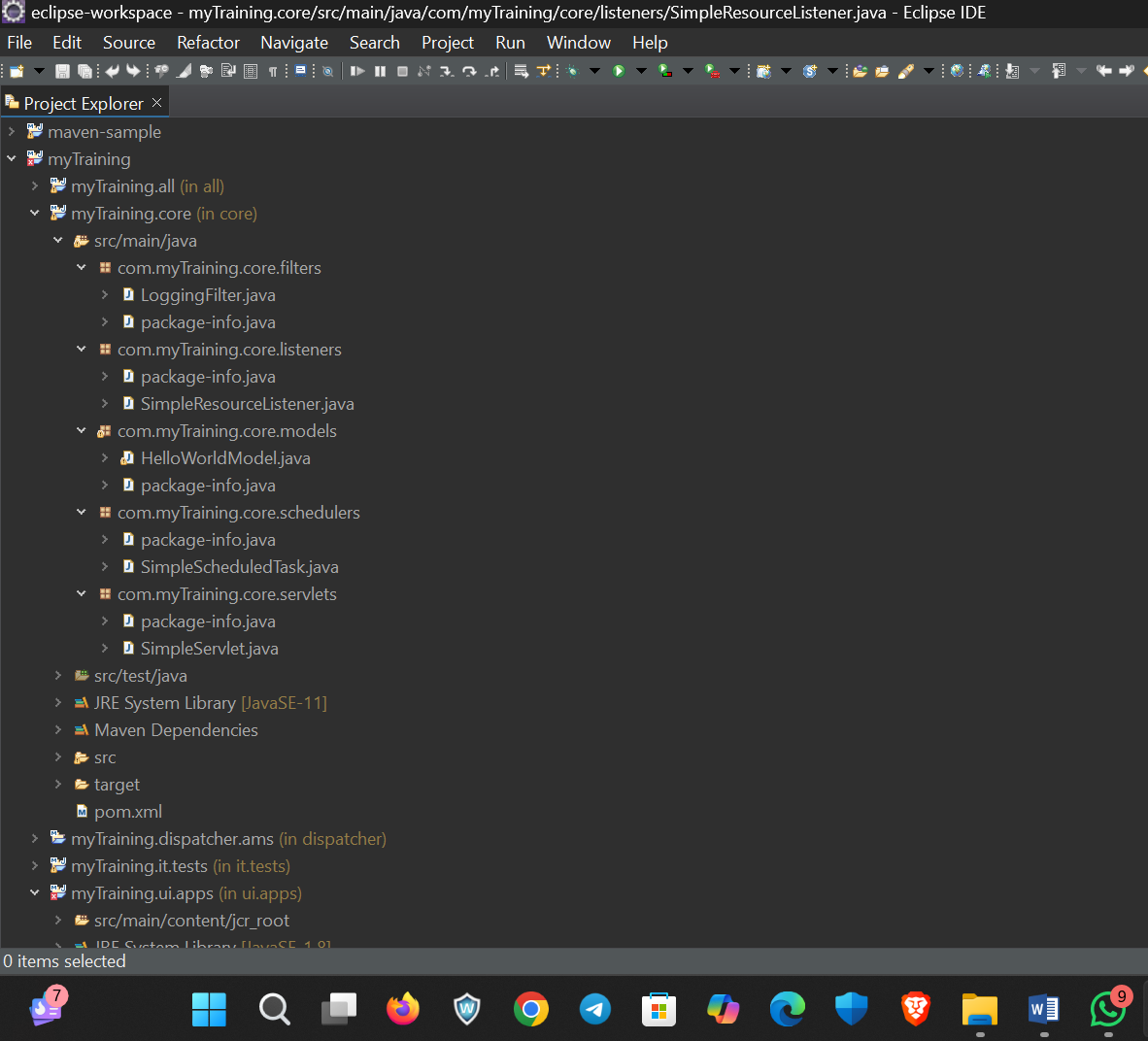
**DAY-02 TASK**

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

The core module in AEM project contains the backend logic for components. It contains java code and module that ensures the data is processed made available to the components in the UI.

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

The core folder in the project contain five different packages which are filters, listeners, models, schedulers and servlets each package has one package-info.java file. for filters package there is LoggingFilter.java file, for listeners package there is SimpleResourceListener.java file , for models package there is HelloWorldModel.java file, for schedulers there is SimpleScheduledTask.java file and for the servlets package there is SimpleServlet.java file



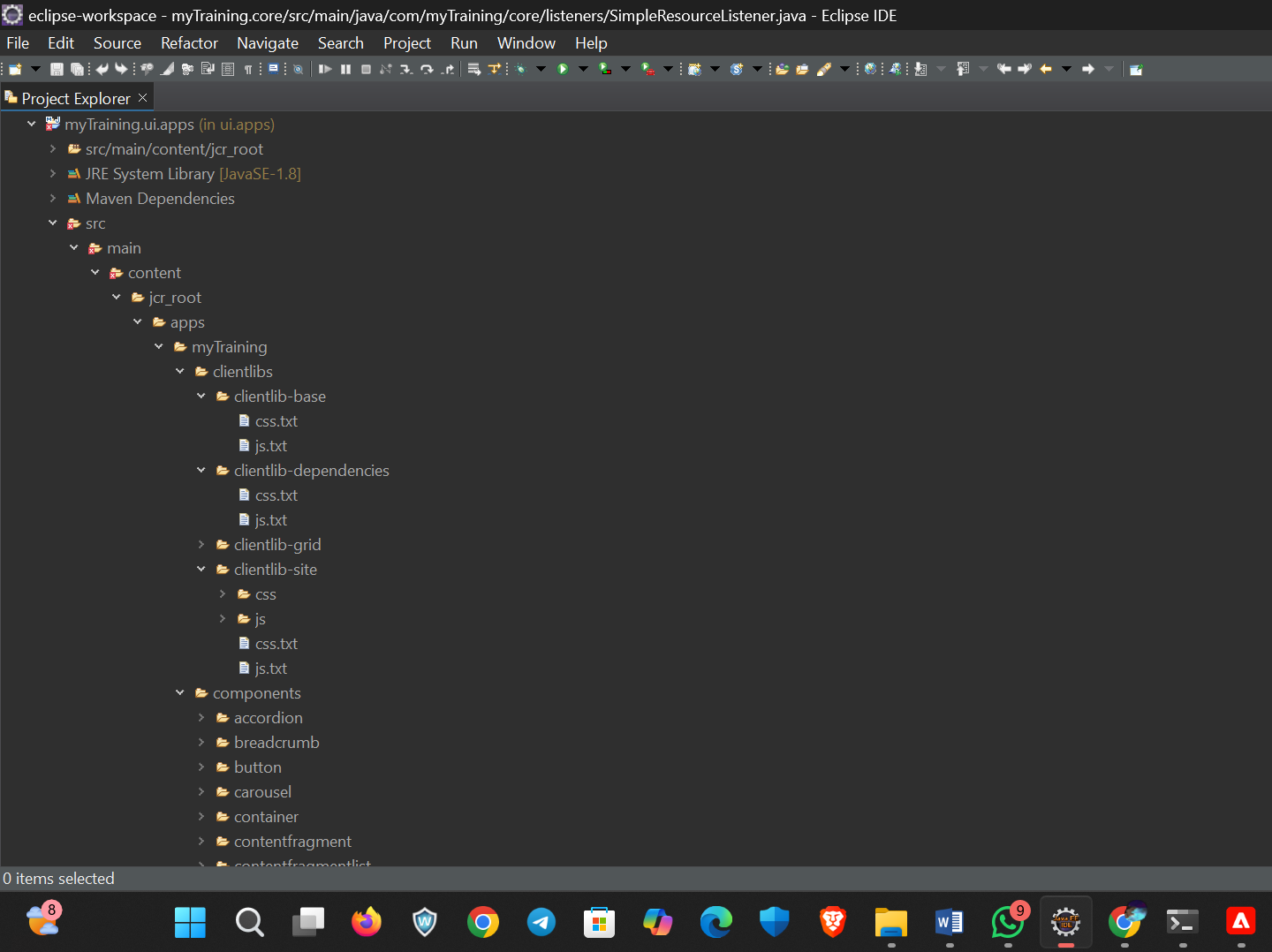
**Explain the role of ui.apps in AEM projects.**

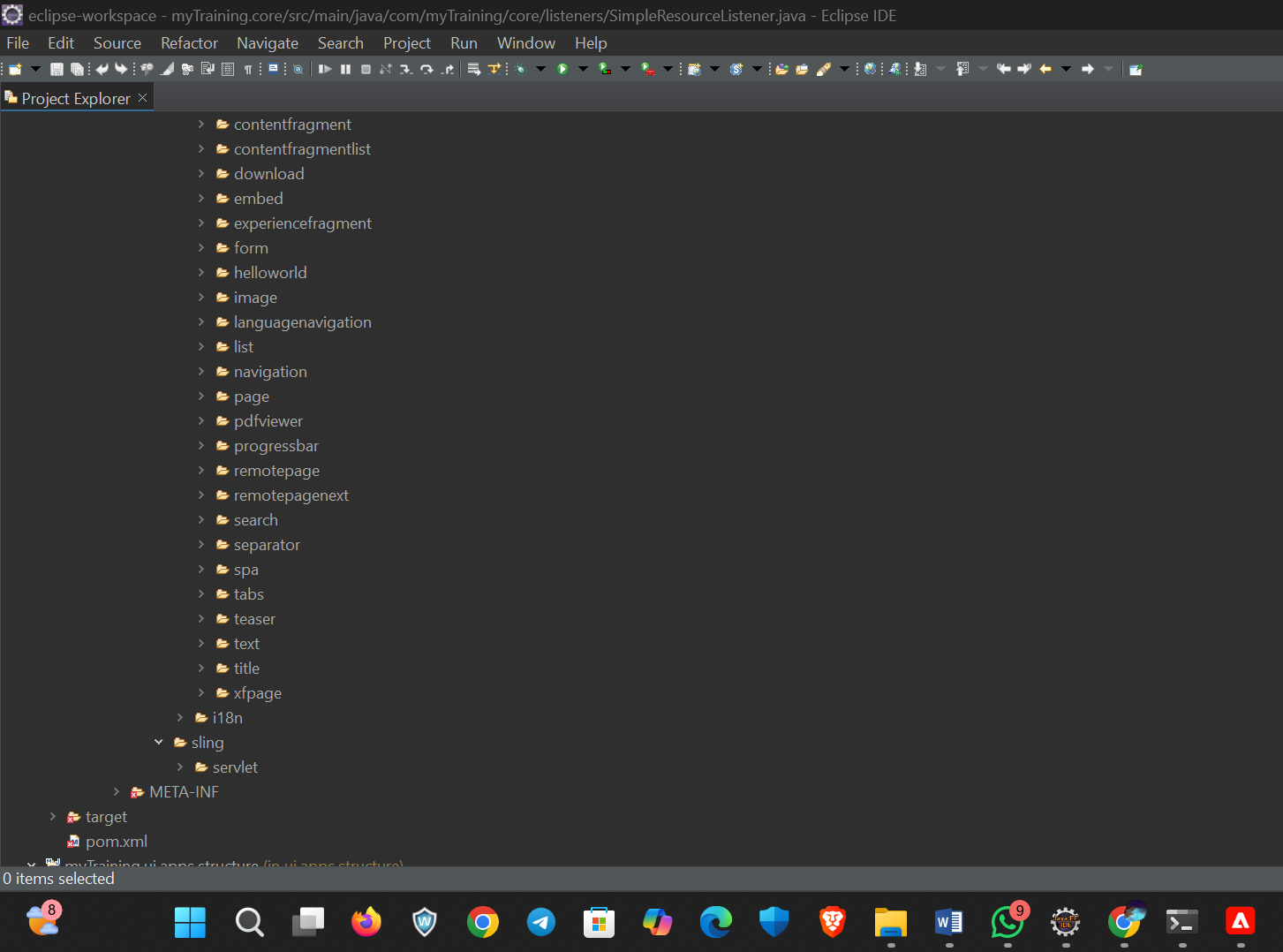
The ui.apps module in AEM project contains the frontend code and configuration for the project.

This module ensures that content is displayes correctly and provide authoring capabilities in AEM.

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

Each component has its own folder.It located in structure like inside ui apps/src/main/apps/components

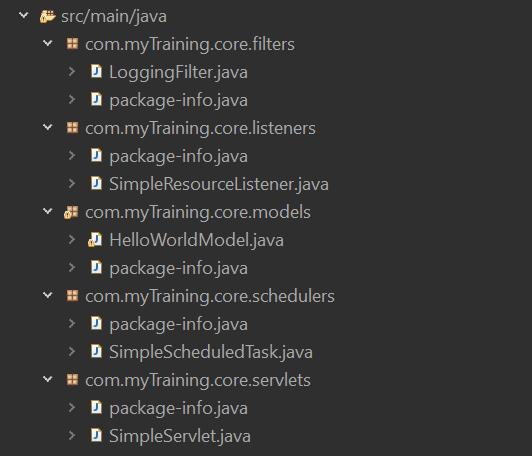
****

****

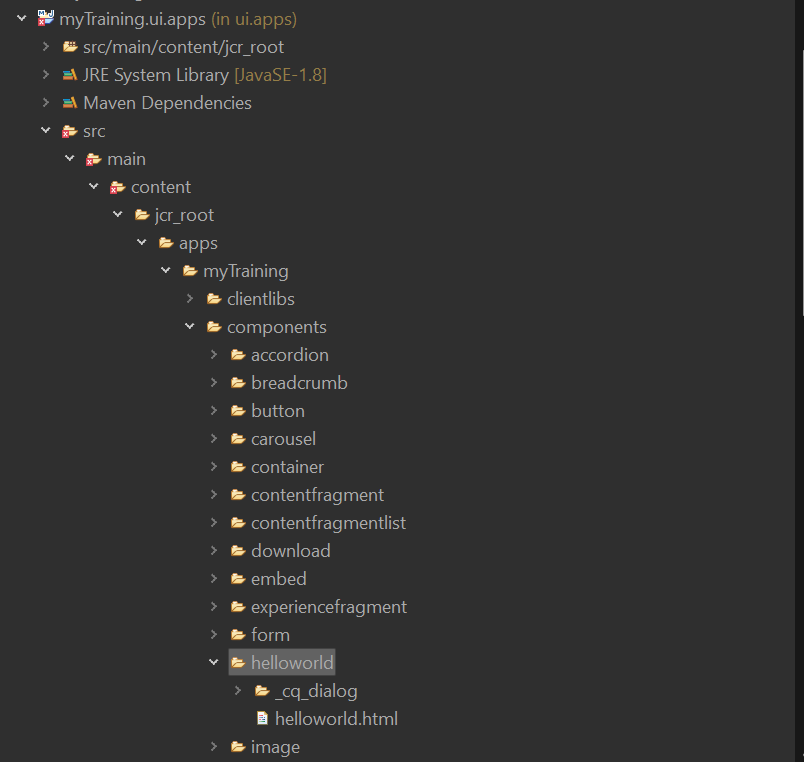
**Hello World Component:**

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

In core The java file is found inside the models package and its paths is like /core/src/main/java/com.myTraining.core.models/HelloWorldModel.java

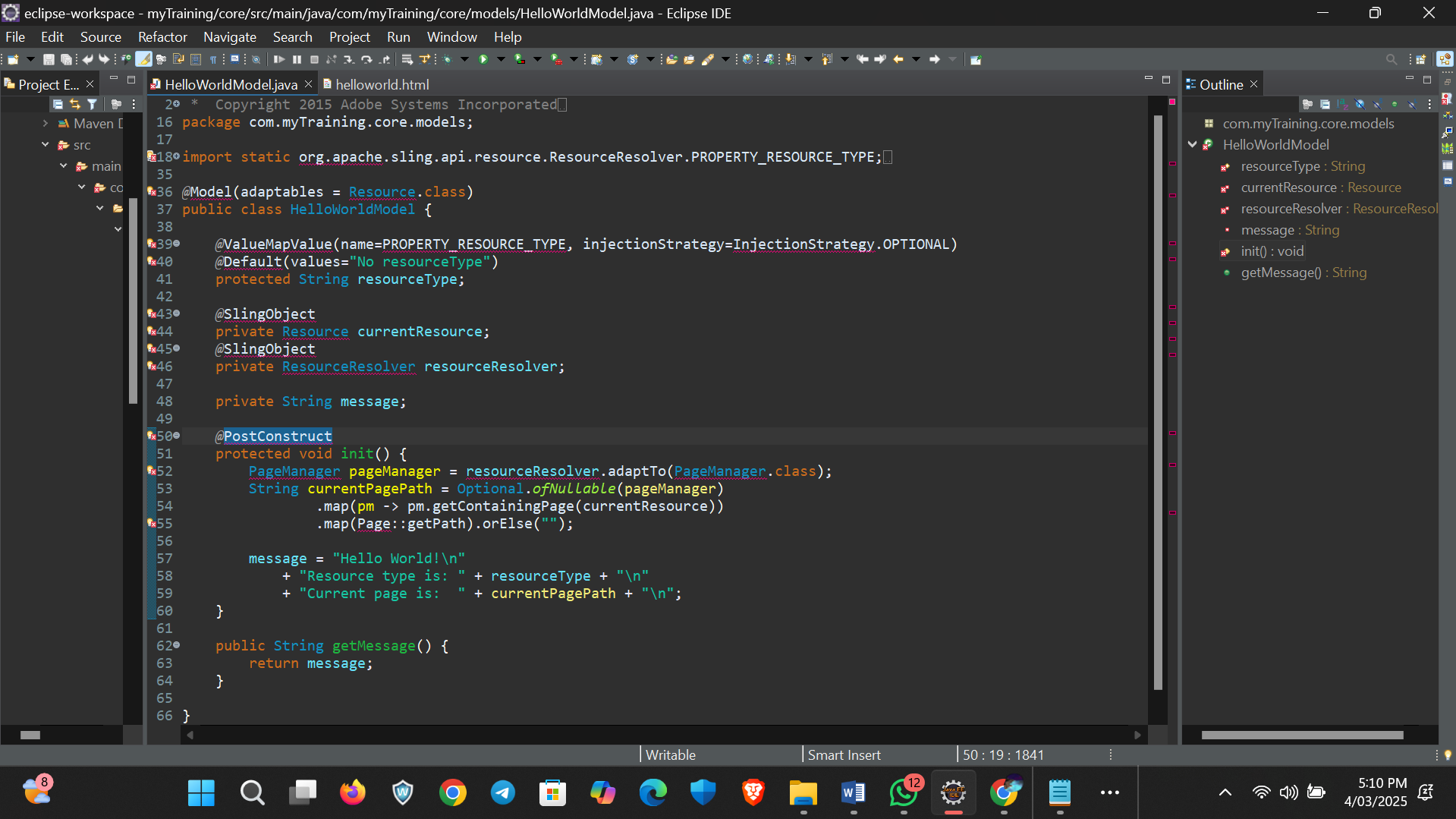


In ui.apps the files are found in components folder and the path is like /src/main/apps/components/helloworld/helloworld.html



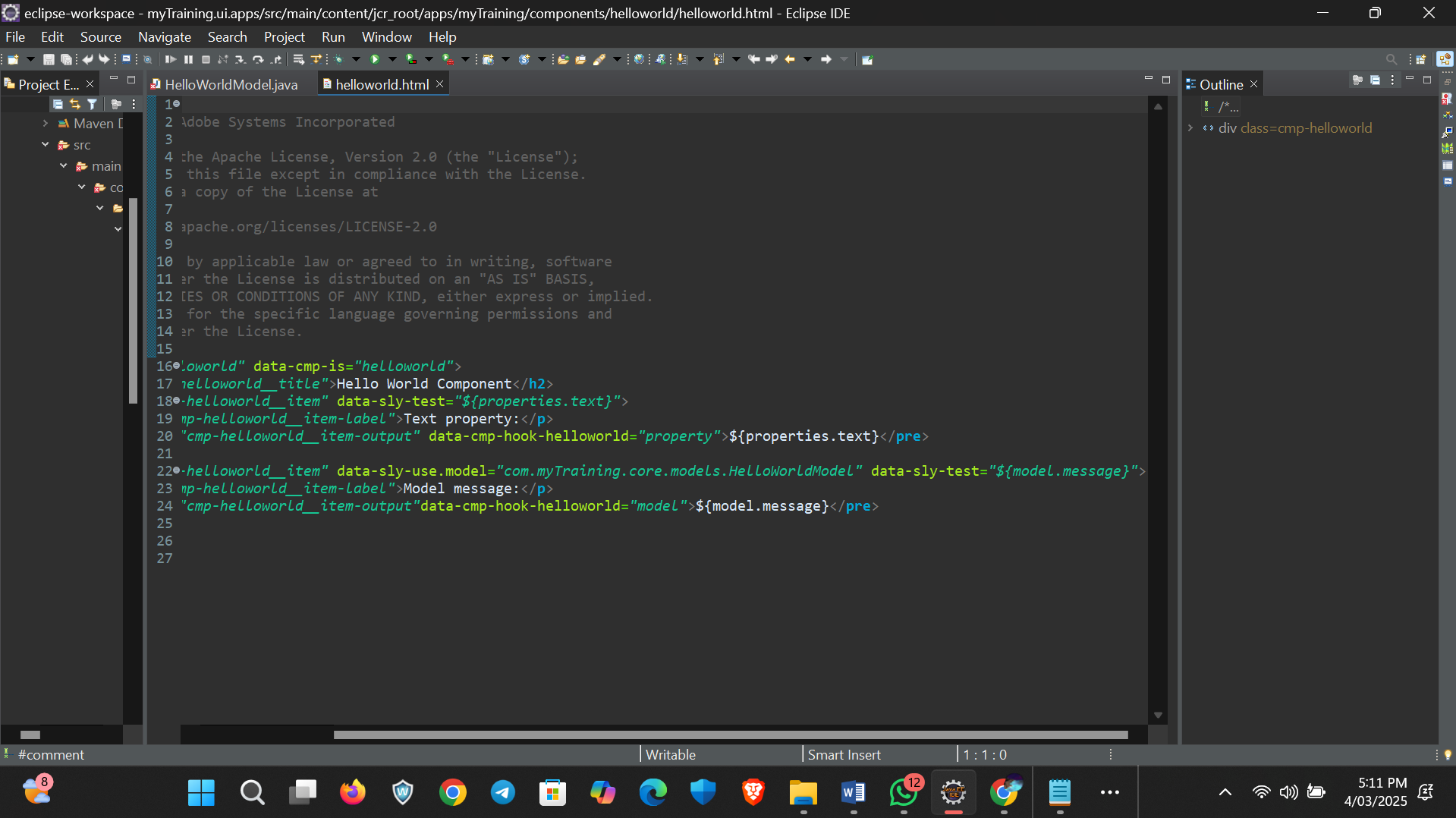
**Explain the Java class (in core) for the Hello World component.**

The java class which is HelloWorldModel.java file is a sling model that retrieves the resource type and current page details. It retrieves the component’s resource typr and find the page it is placed on. The message variable is used in HTL to display data. @Model(adaptables = Resource.class) defines the class as a Sling Model.



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

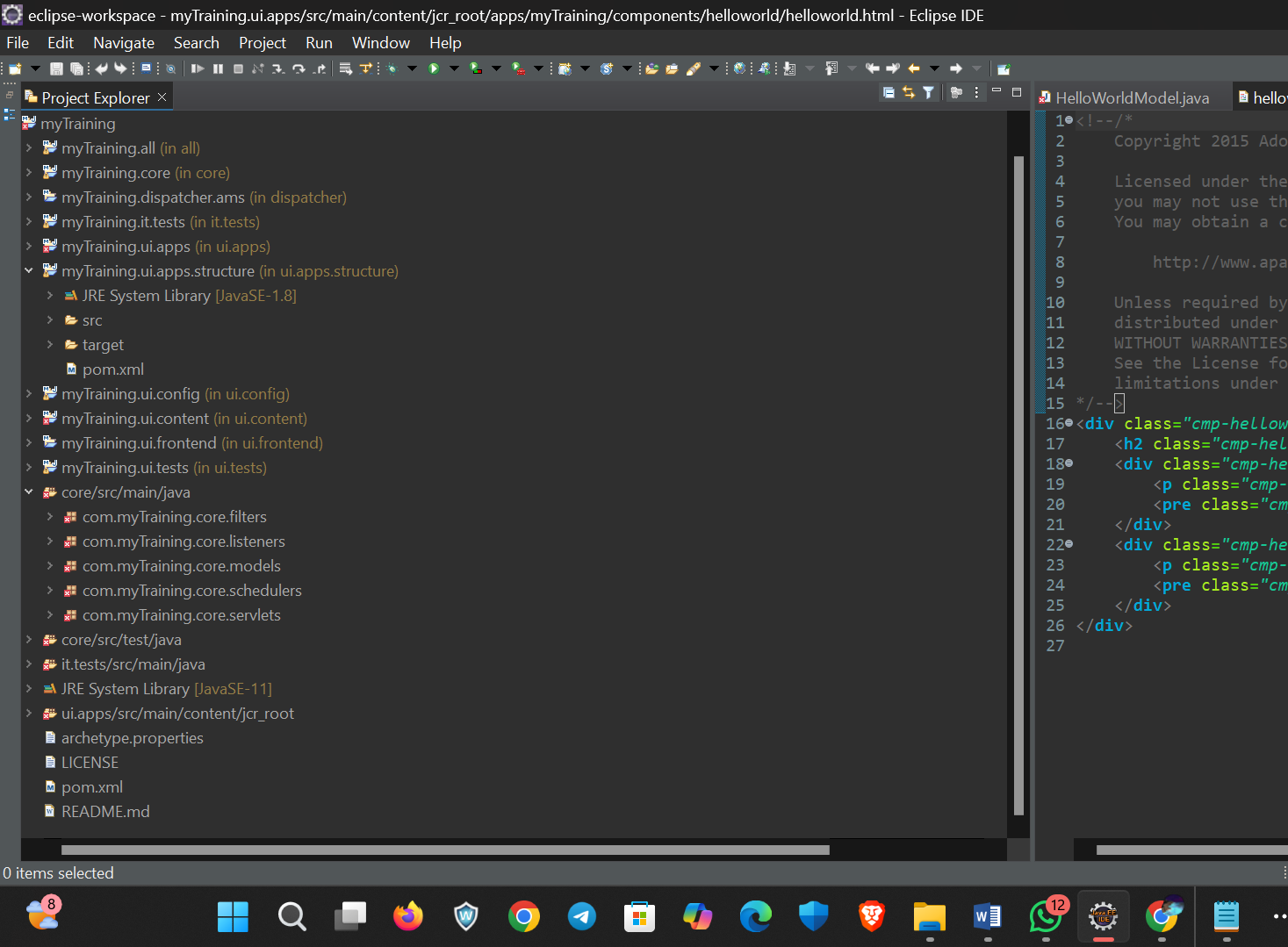
The HTL file in component displays data from java model ${properties.text} fetches values stored in the component’s properties. ${model.message} retrieves the processed message from the HelloWorldModel Sling Model.



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

In AEM project, properties and dialogs for a component are defined in the \_cq\_dialog This file is used to create the authoring dialog in the AEM Touch UI, allowing content authors to enter data for the component the file located in ui.apps/src/main/content/jcr\_root/components/helloworld/\_cq\_dialog.xml

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

****

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

To build maven project in AEM first need create a maven project we created the maven project with command Prompt run as administrator with command as “mvn -B org.apache.maven.plugins:maven-archetype-plugin:3.2.1:generate -D archetypeGroupId=com.adobe.aem -D archetypeArtifactId=aem-project-archetype -D archetypeVersion=39 -D appTitle="myTraining" -D appId="myTraining" -D groupId="com.myTraining" -D aemVersion=6.5.5 -D includeExamples=n -D includeErrorHandler=y” after the maven build project is success import the project in ide where I use Eclipse ide after importing right click the parent folder and go to properties which contain all other and select under maven there is project facet click covert to facet and select java 11 version and do the same for core.After this process success goes to the maven project and open the terminal give this command to maven build in AEM “mvn clean install -PautoInstallPackage -Dmaven.test.skip=true”

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

In AEM project, dependencies are managed using maven in the pom.xml file.Dependencies allow the project to use external libraries, AEM APIs and custom modules.

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

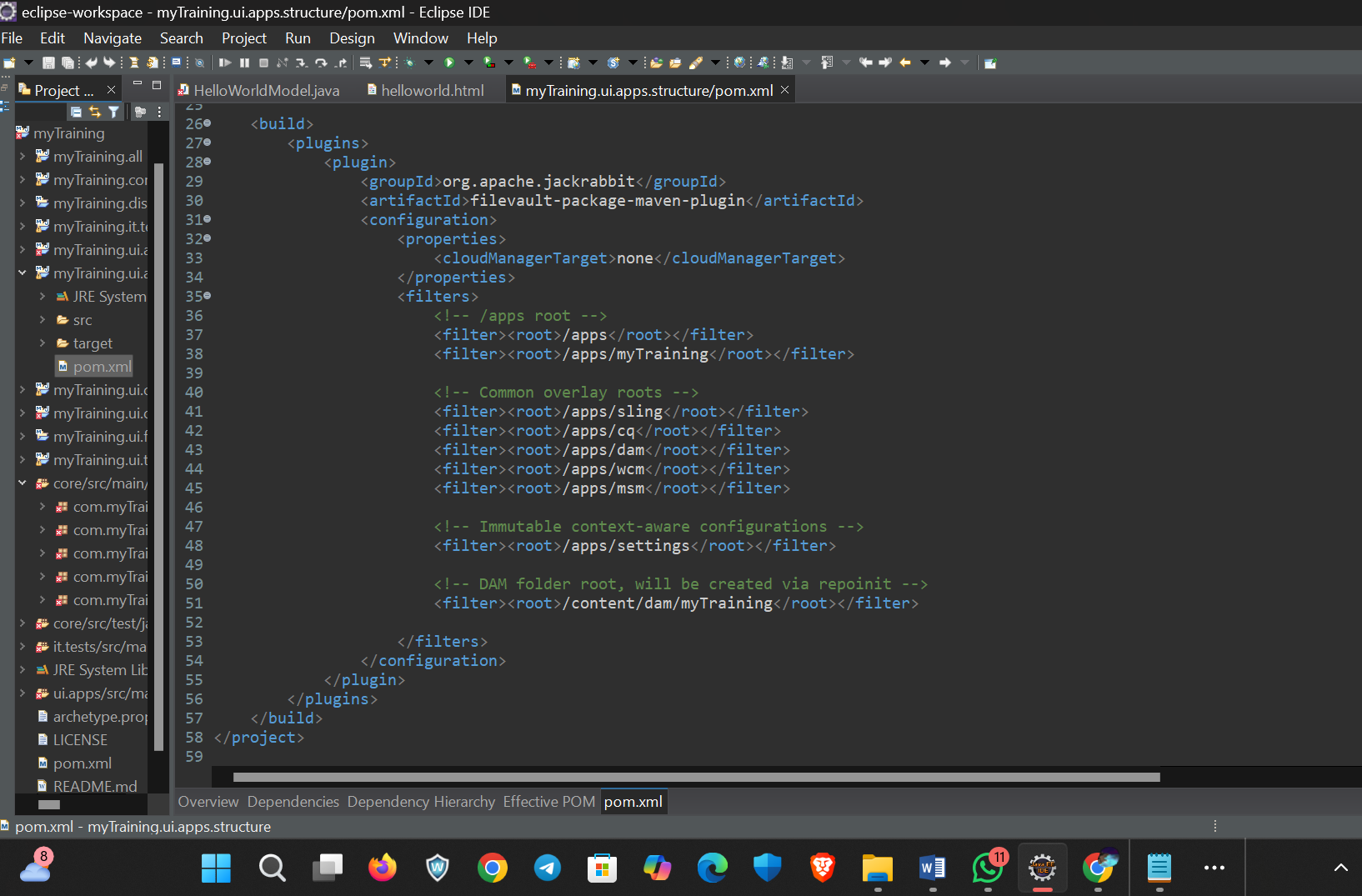
Maven is preferred over other build tools in AEM because of its powerful dependency management, automation and integration with AEM’s structure. It automatically downloads and manages dependencies through thepom.xml.No need to manually include JAR files. This makes it easy to organize modules like core, ui,apps and ui.content.Maven provides build automation with commands like mvn clean install to compile, tests, packages and deploys in a single step.

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

AEM projects follow the Adobe Archetype, which is Maven-based. This makes it easy to organize modules like core, ui,apps and ui.content. It manages libraries automatically through pom.xml. It downloads dependencies from Adobe’s Maven repository, eliminating manual JAR file management.

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

Maven simplifies dependency and plugin management in AEM projects through the pom.xml file. Dependencies are **external libraries.** Maven automatically downloads required **JAR files** from Adobe’s and Apache’s repositories.

****

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

The mvn clean install command is one of the most commonly used Maven commands in AEM development. It builds, compiles, tests, packages, and installs the project.

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

Maven allows developers to build and deploy AEM packages directly to an AEM instance . This is done using Maven profiles and the FileVault Package Plugin. To build and deploy the package to local AEM instance we use: mvn clean install -PautoInstallPackage -Dmaven.test.skip=true

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

Maven profiles in AEM allow developers to build and deploy specific parts of a project efficiently. The most commonly used profiles are: autoInstallPackage which deploys the entire AEM . we can check in http://localhost:4502/crx/packmgr/index.jsp

autoInstallBundle which Deploys **only the backend OSGi bundle** to AEM without affecting the frontend.package. we can check in http://localhost:4502/system/console/bundles

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

The dumplibs tool in AEM is used to debug and inspect client libraries (clientlibs). It helps developers understand how CSS, JavaScript, and other front-end resources are being resolved and loaded in AEM**.**

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

The dumplibs tool in AEM provides a way to inspect, debug, and analyze client libraries (clientlibs), including their dependencies and loading order. To open the dumplibs page in browser: http://localhost:4502/libs/granite/ui/content/dumplibs.html

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

Client Libraries (clientlibs) in AEM are used to manage and organize CSS, JavaScript, and other static assets efficiently. They help in modularizing frontend resources for better performance and maintainability.

