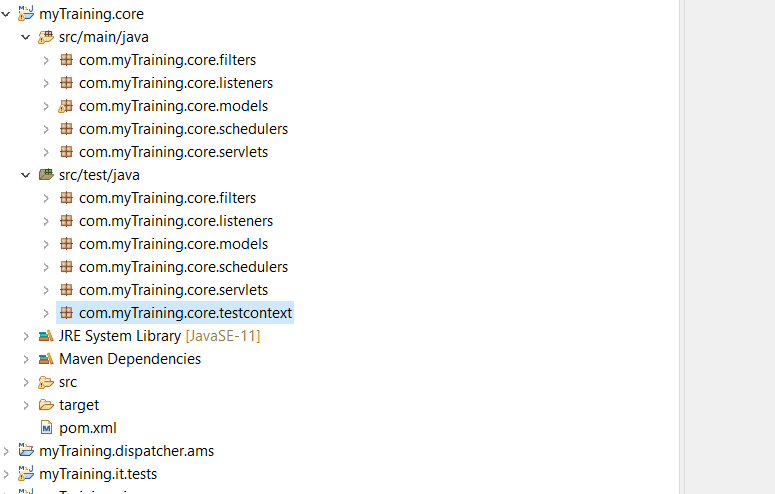
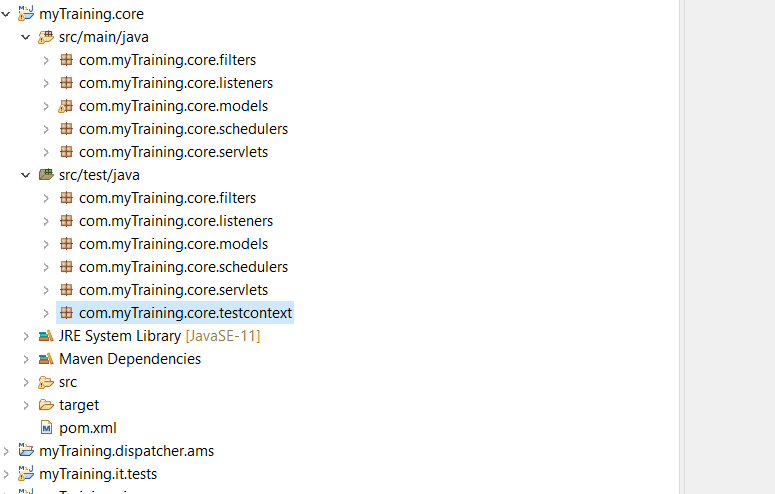
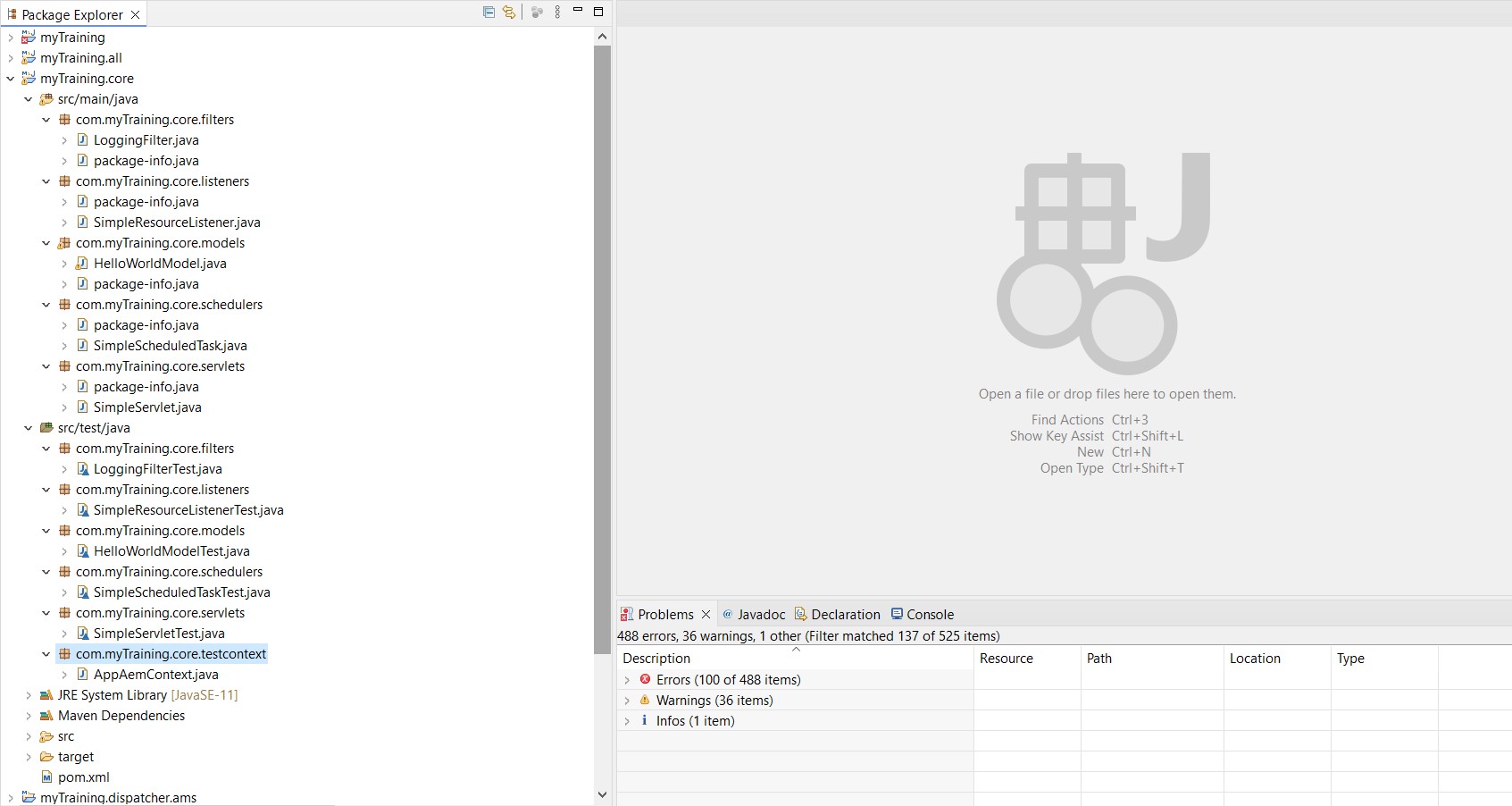
1.What is the purpose of the core module in AEM?

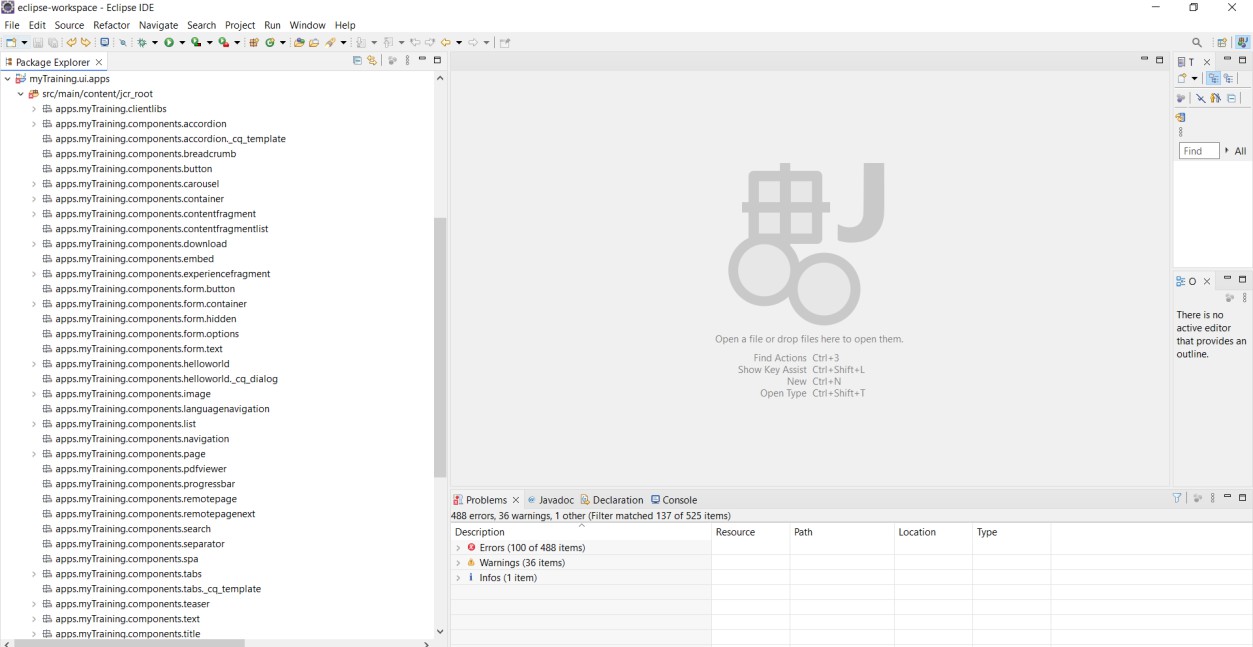
- The core module in AEM provides a set of pre-built and reusable components for development. It contains key components such as page templates, content elements, and site utilities. This module helps developers in minimizing custom coding.

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

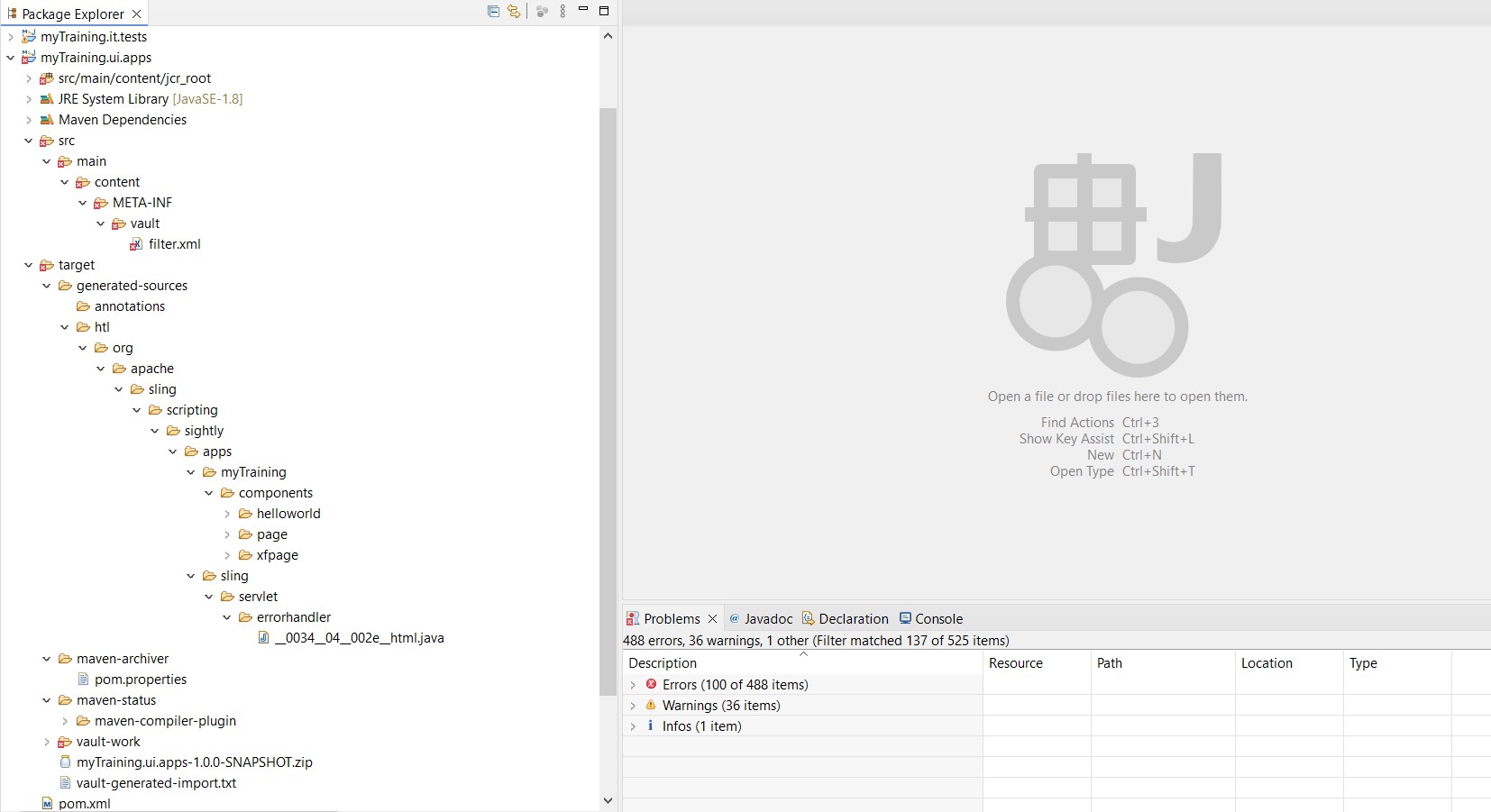
- The root folder of an AEM project is mostly made up of backend logic and business rules. It contains the files and codes like Java Classes, POM File, Test Cases, META-INF File ,etc.

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

- In Adobe Experience Manager (AEM) projects, the ui.apps module handles and deploys frontend resources such as components, templates, dialogs, and client-side codes. Client libraries  in ui.apps enhance frontend performance through the efficient handling of CSS and JavaScript.



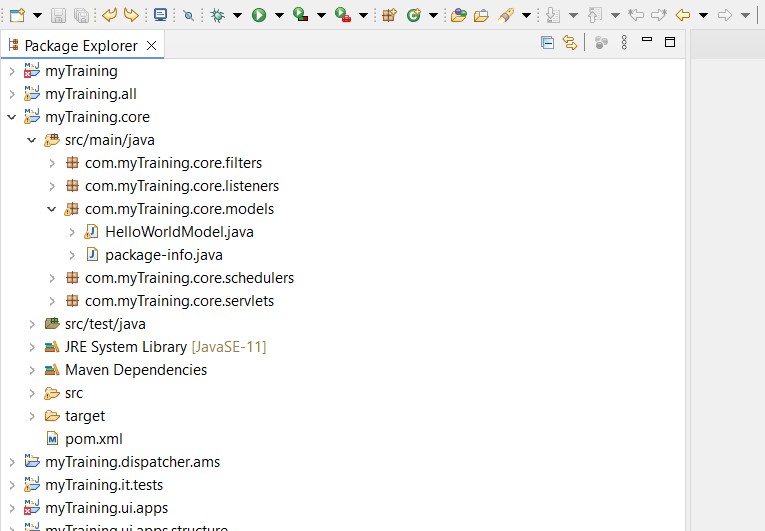
4.How are components structured in the ui.apps folder?

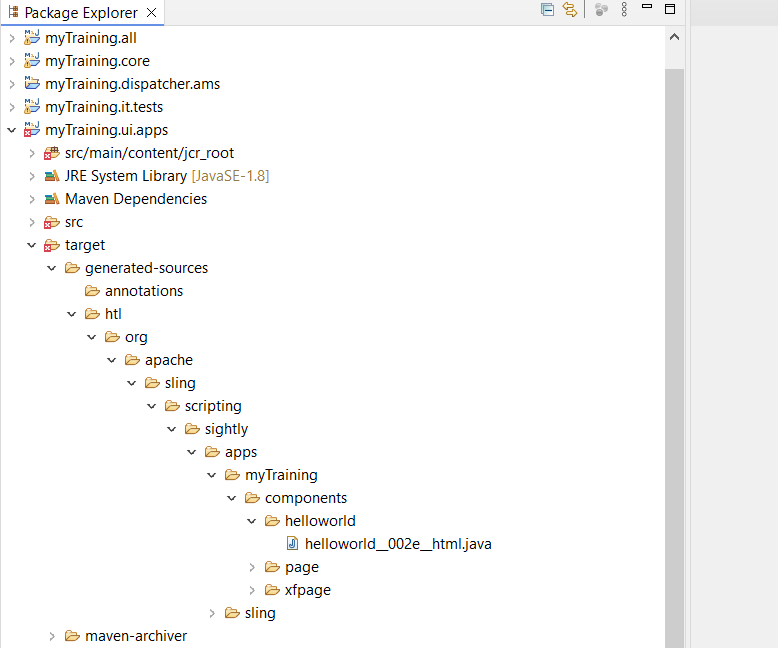
- In AEM, components are kept in the ui.apps folder under /apps/<projectname>/components/. Each component gets its own folder, like /apps/myproject/components/hero-banner. Inside these folders, you'll find HTL scripts. These scripts are used to make the component appear on the website. There is also a cq:dialog file, which is important for setting up the component's options and properties when using the touch based UI.

**5.Hello World Component:**

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

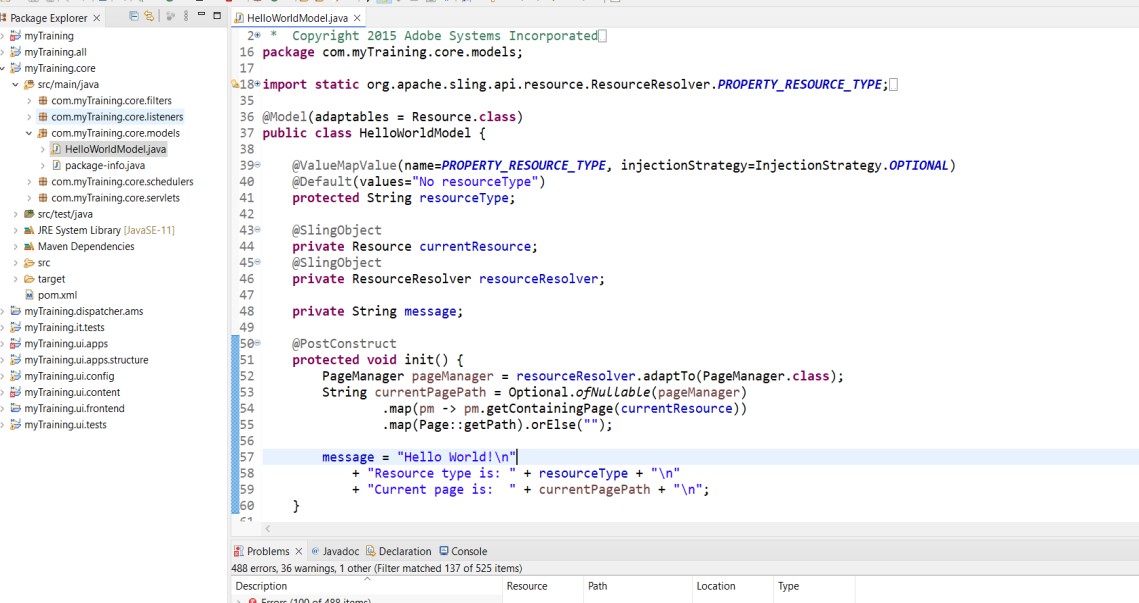
- In Core Module- /core/src/main/java/com/myproject/core/models/HelloWorldModel.java



- in ui.apps- /ui.apps/src/main/content/jcr\_root/apps/myproject/components/helloworld

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

- The HelloWorldModel.java class is a type of program known as a Sling model. Its primary function is to retrieve and process data for a web page template called HTL.It is designed to prepare the correct data for a website.



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

- The HTL script in the ui.apps section dynamically creates the component's display using data from the Sling Model in the core. This keeps the logic separated from what you see on the screen. The script collects data from the Sling Model and uses expressions to display this content.

* How are properties and dialogs defined for this component?

- In AEM, "cq:dialog" sets up properties and dialogs so that content creators can configure components using the Touch UI. These configurations are stored in the Java Content Repository and accessed by the component as needed.

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

- Core modules,

- ui.apps

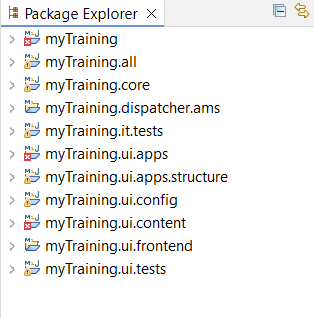
- ui.content

- ui.config

- it.tests

- all

- dispatcher, etc.



7.How does Maven build these modules?

- Maven is used to build software like AME by dividing it into modules, each with its own task. A  main file (pom.xml), oversees managing all these parts. During the AEM build process, maven compiles the code, converts it into a format that computers can use, then packages it for deployment. Finally, it deploys the software to AEM, ensuring all components function together.



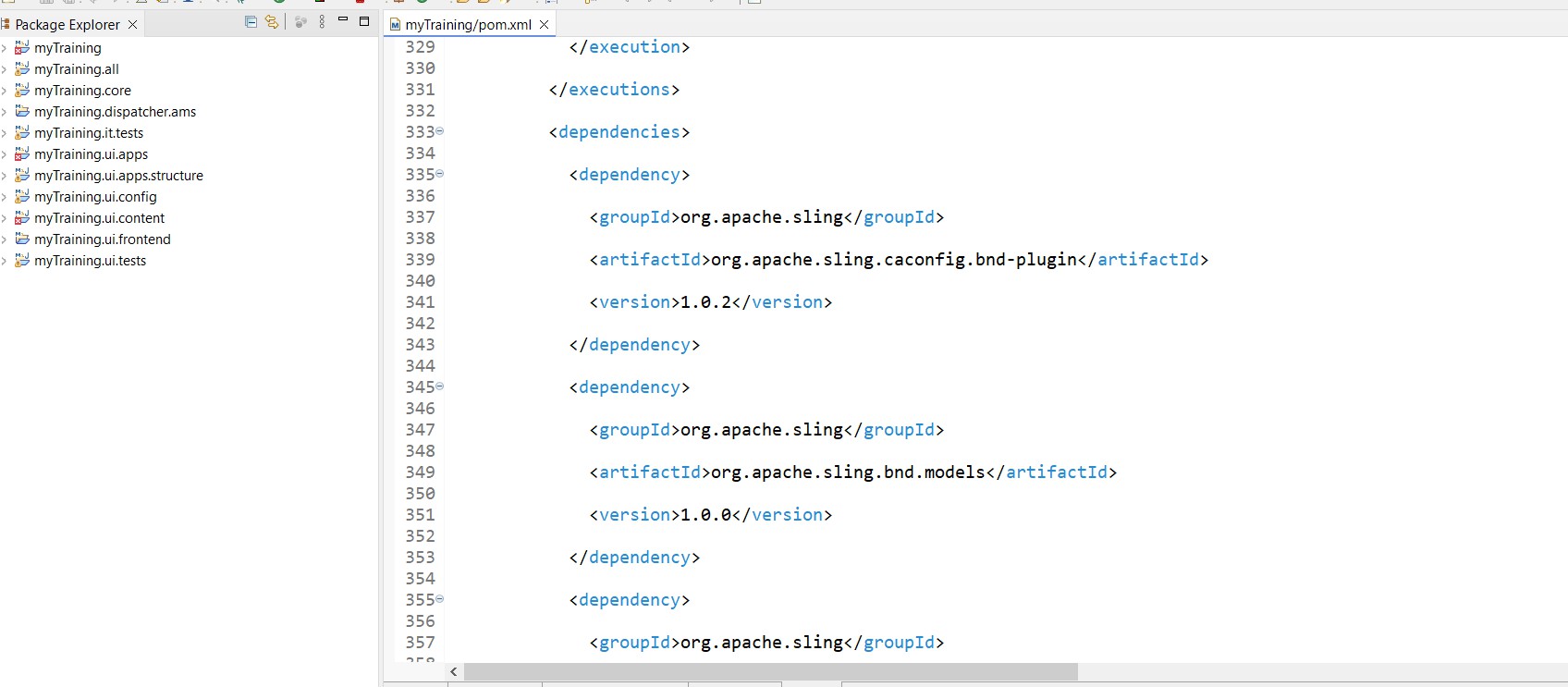
8.Explain the build lifecycle of Maven in the context of AEM.

- Maven automates the process of compiling, packaging, and deploying these sections in the proper order using plugins and dependencies. This organized method ensures that all parts work together, minimizes mistakes, speeds up the process, and finalizes the product without needing manual handling.

* **Clean Phase**
* **Validate Phase**
* **Compile Phase**
* **Test Phase**
* **Package Phase**
* **Verify Phase**
* **Install Phase**
* **Deploy Phase**

9.How are dependencies managed in pom.xml?

- In Maven, you handle dependencies in the pom.xml file, particularly in the section. This section specifies the external libraries needed for your AEM project, such as OSGi bundles, AEM APIs, and testing frameworks. Maven takes care of downloading and including these libraries in your project build automatically, eliminating the need for manual downloads.



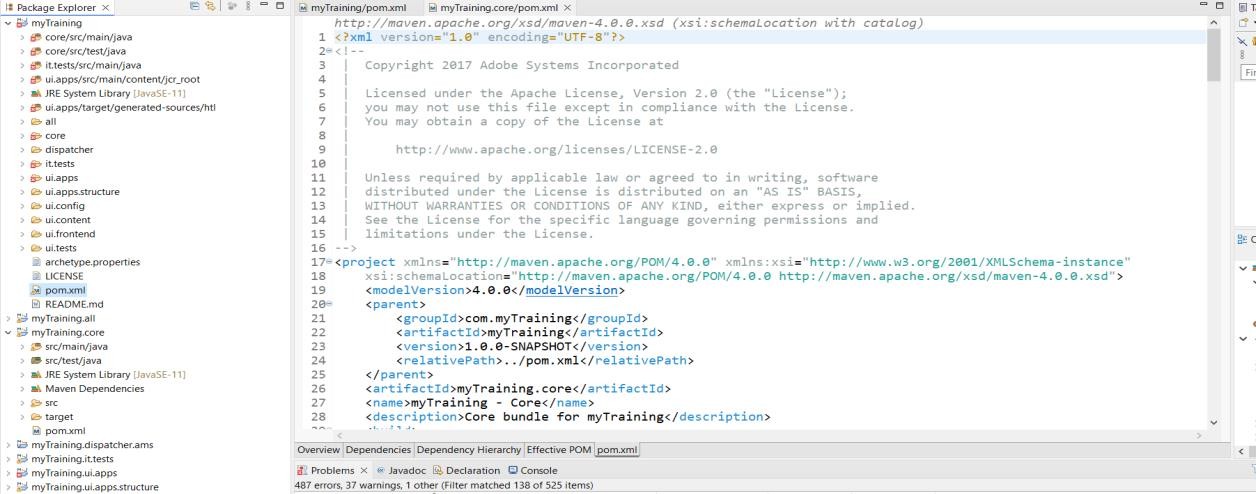
10.Why is Maven used instead of other build tools?

- Maven is widely used over other build tools because it makes managing dependencies easier and provides a clear process for building projects. By handling libraries, Maven ensures that everything stays consistent, no matter where you're working, and it connects easily with repositories for efficient code management. With a strong community and widespread use, maven remains a dependable and effective tool for handling complex projects.

11.What advantages does Maven offer for AEM development?

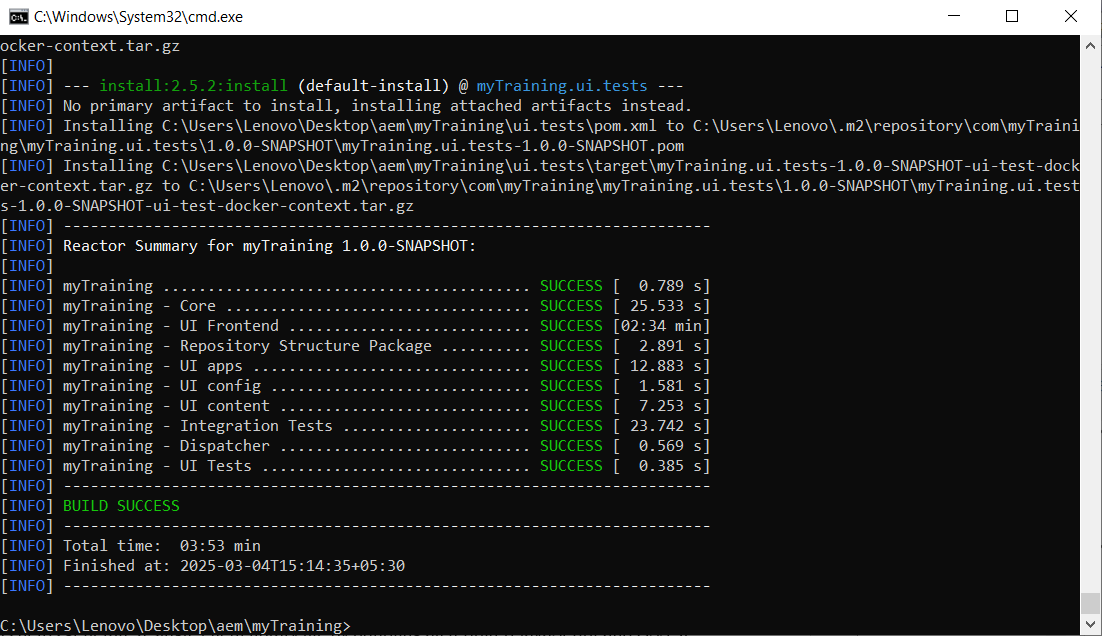
* Easily handles AEM APIs, OSGi bundles, and third-party libraries.
* Ensures a structured process for compiling, testing, packaging, and deployment.
* Efficiently manages AEM modules like core, ui.apps, and ui.content.
* Works well with Adobe’s repository and CI/CD tools for automation.
* Reduces manual setup and simplifies project management.

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

- Maven is a tool used in AEM projects to manage software components. It automates the process of finding, updating, and integrating the necessary libraries and tools. Maven also helps prevent issues with different software versions through its feature, which carefully tracks and manages versions to ensure all parts work together seamlessly.

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

- The "mvn clean install" command in an AEM project accomplishes these tasks:

* Removes previously compiled files and clears target folders.
* Compiles Java code, forms OSGi bundles and packages AEM components, templates, and content into ZIP files for deployment.
* Conducts unit and integration tests to confirm the code's readiness for use.
* Stores the completed JAR and ZIP files in the local Maven repository for future use.
* Produces a comprehensive AEM package (all.zip) that is ready to be deployed on an AEM instance.

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

* **Deploy the full project:**

mvn clean install -PautoInstallPackage

* **Deploy only the OSGi bundle (core module):**

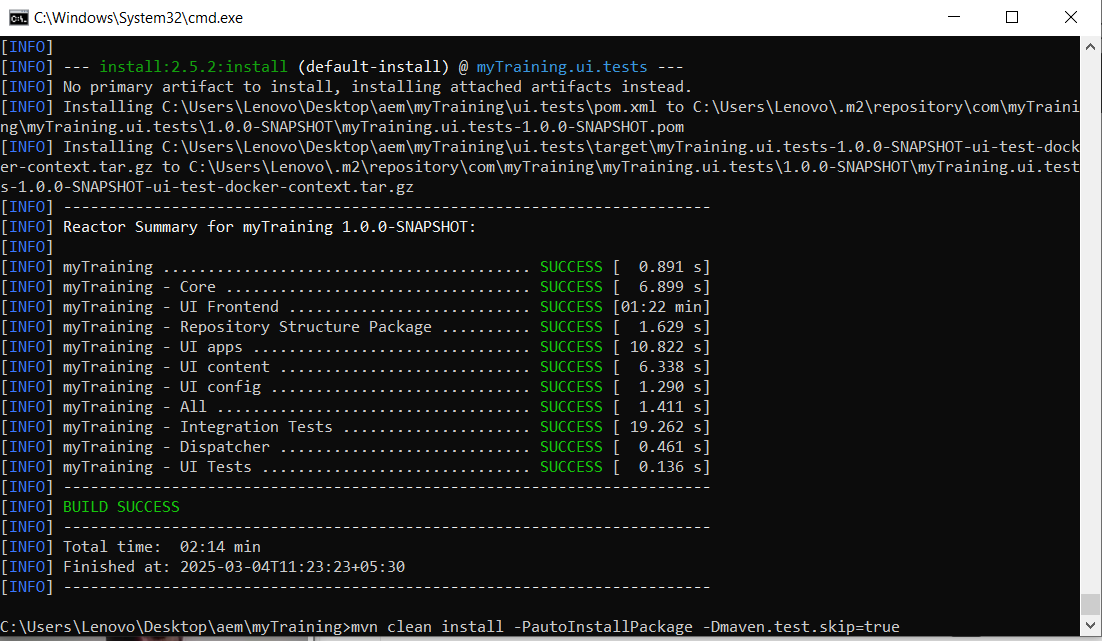
mvn clean install -PautoInstallBundle

* **Deploy to a specific AEM instance:**

mvn clean install -PautoInstallPackage -Daem.host=<AEM\_HOST> -Daem.port=<PORT>

* **Deploy only the ui.apps module:**

mvn clean install -PautoInstallPackage -pl ui.apps



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

* **utoInstallPackage** – Builds and installs the full AEM content package (all.zip) into the local AEM instance, including components, templates, and configurations.

mvn clean install -PautoInstallPackage

* **autoInstallBundle** – Deploys only the core module without affecting other parts of the project, making it useful for backend code updates.

mvn clean install -PautoInstallBundle

16.What is the purpose of dumplibs in AEM?

- The dumplibs feature in AEM is a tool for checking and solving problems with client libraries. It lists all the JavaScript and CSS files on a specific page. This helps developers spot any resources that are missing, duplicated, or not loaded correctly on the client side.

17.How can you view client libraries using dumplibs?

* **Open the Client Library Debugging Tool:**

http://localhost:4502/libs/granite/ui/content/dumplibs.html

* **View Client Libraries for a Specific Page:**

http://localhost:4502/libs/granite/ui/content/dumplibs.html?path=/content/mysite

* **Enable Debug Mode to See Merged Clientlibs:**

http://localhost:4502/libs/granite/ui/content/dumplibs.html?debug=true

18.Explain how client libraries are structured in AEM.

- In AEM, Client Libraries known as clientlibs, are used to efficiently manage front-end assets, which include CSS files, JavaScript, fonts, and images. These are organized under the ui.apps module and are stored in a specific JCR path.Clientlibs are included in AEM components using HTL or JavaScript APIs for efficient asset management.

 /apps/<projectname>/clientlibs

