**AEM PROJECT STRUCTURE**

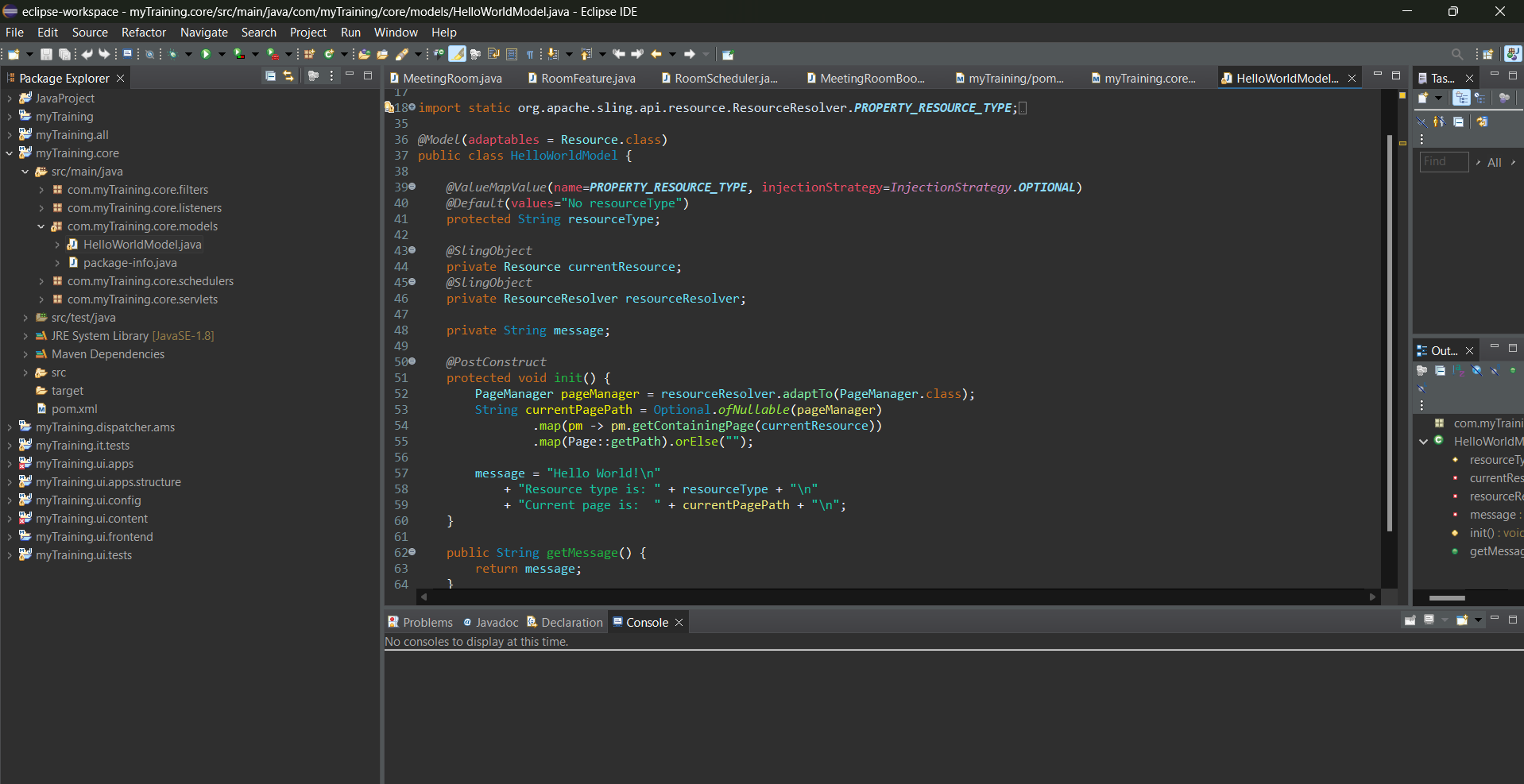
**MAVEN PROJECT**

**1.Project Structure of core and ui.app :**

* core: Contains Java classes for AEM services, models, and OSGi components.
* ui.apps: Contains frontend components (HTL templates, dialogs, client libraries).

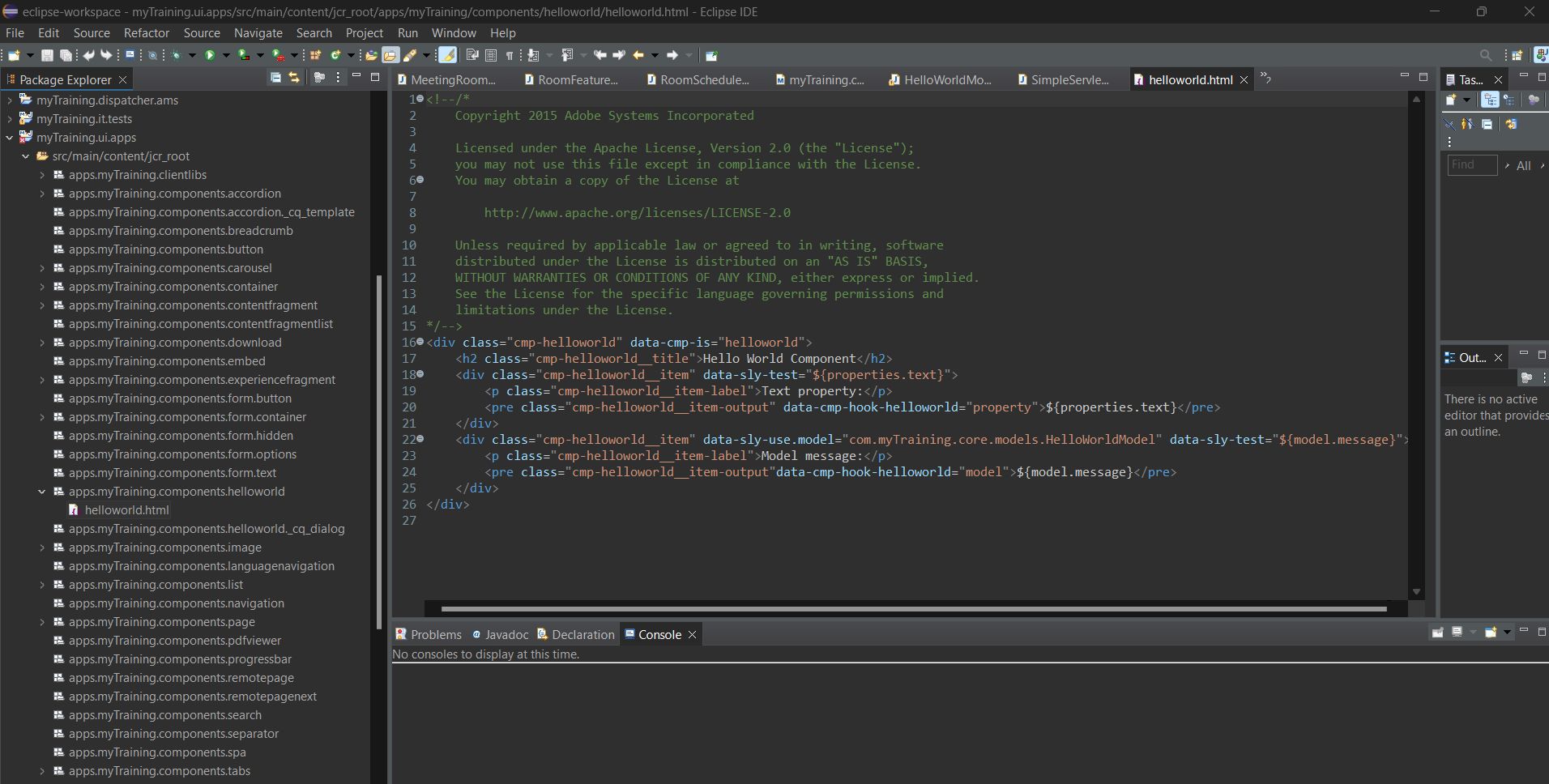
**Core Module:**

**Hello World Component**

****

**Ui.apps**

**Hello World .html**

****

**2. How AEM modules are build ?**

* AEM projects use Apache Maven to manage dependencies and build modules.
* Modules are defined in POM files (pom.xml).
* AEM projects typically follow a multi-module structure (core, ui.apps, ui.content, etc.).

Commands:

1. Builds all modules:

mvn clean install

1. Deploys the package to AEM:

mvn -PautoInstallSinglePackage install

**3.Why Maven?**

* **Dependency Management** → Automatically resolves dependencies (like Sling, OSGi, etc.).
* **Module** Structure → Allows separate development of core logic (core), frontend (ui.apps), and content (ui.content).
* **Automation** → Enables CI/CD integration for automated builds and deployments.

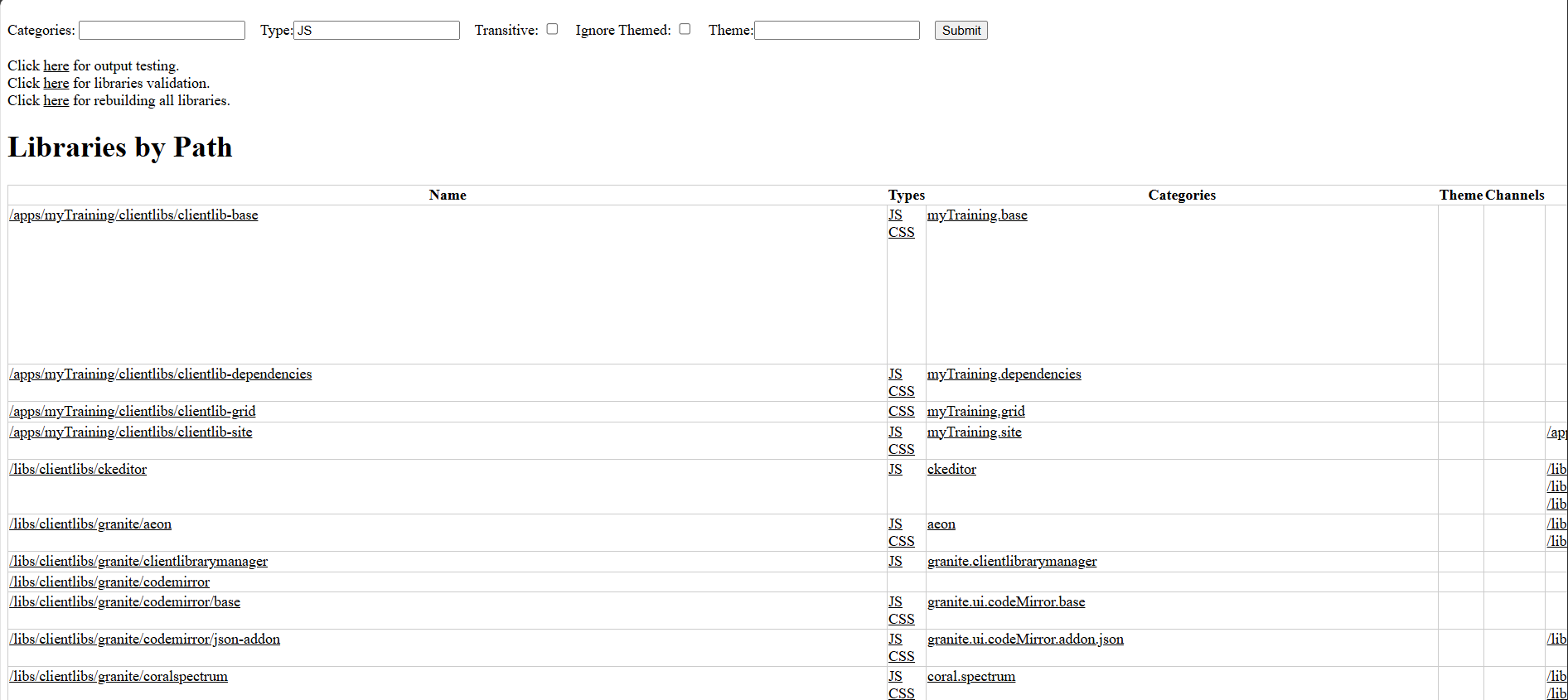
**4.How we build AEM project using different Commands?**

* Builds all modules: mvn clean install
* Deploys the full package to AEM: mvn -PautoInstallSinglePackage install
* Installs individual modules: mvn -PautoInstallPackage install
* Installs only the core bundle: mvn -PautoInstallBundle install
* To deploy it to a publish instance: mvn clean install-

PautoInstallSinglePackagePublish

**5.Role of dumplibs for client libs?**

dumplibs.html is a debugging tool in AEM used to inspect Client Libraries (ClientLibs).



Clientlib-site:

