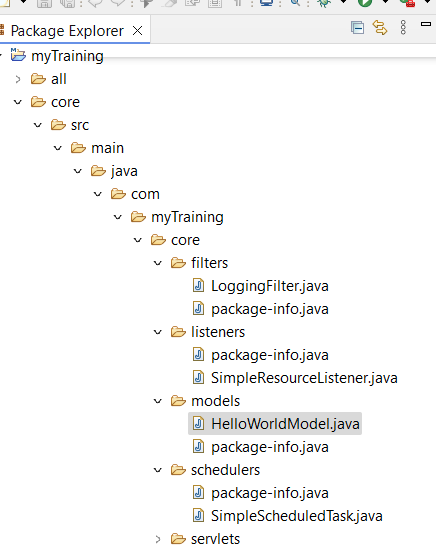
**Day – 2**

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

The **core module** in AEM contains the backend logic, written in Java, that powers AEM components. It includes Java models, OSGi services, and business logic used across the project.

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

The **core folder** contains folders like src, main, filters, listeners, models, schedulers as we can see in image

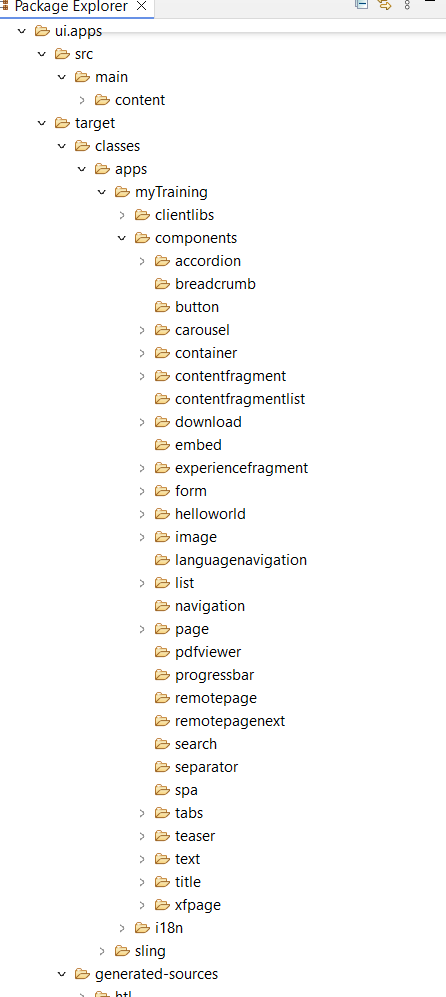


**3.Explain the role of ui.apps in AEM projects.**

The **ui.apps** module contains front-end resources such as components, templates, client libraries, and configurations. It is responsible for defining how content is presented in AEM.

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

The Components are structured in a semantic order as we can see below



**5.Hello World Component:**

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

The HelloWorld is located as java file in core as core>models>HelloWorldModel.java and as component in ui.apps

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

The Java class (HelloWorldModel.java) is a Sling Model annotated with @Model(adaptables = Resource.class), exposing methods like getMessage() to return text.

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

The HTL (component.html) uses:

<sly data-sly-use.model="com.example.models.HelloWorldModel">

${model.message}

</sly>

It binds the Java Model to frontend



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

AEM uses Touch UI dialogs to allow users or creators to configure components in the page editor. The dialog is defined in:

/apps/project/components/helloworld/\_cq\_dialog.xml

\_cq\_dialog.xml - defines the AEM dialog fields.

cq:editConfig.xml - configures edit behaviors.

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

core: used for backend logic (Java, OSGi services).

ui.apps: used for components, templates, and configurations.

ui.content: used as sample content for testing.

all: used as maven module for assembling the package.

ui.config – Contains OSGi configurations in a structured format.

dispatcher – Defines Dispatcher configurations for caching and security.

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

Maven builds AEM modules using multi-module projects. Each module (core, ui.apps, ui.content) is compiled and packaged into a deployable AEM package.

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

clean – deletes previous builds.

validate – checks project structure.

compile – compiles Java code.

package – bundles JAR/WCM package.

install – installs to local repository.

deploy – deploys to AEM instance.

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

Dependencies are defined under <dependencies> in pom.xml, with versions controlled via dependencyManagement.

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

Maven automates builds, manages dependencies, and provides a structured way to handle AEM projects.

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

Standardized build process.

Dependency and plugin management.

Multi-module support.

Easy deployment to AEM.

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

Dependencies are declared in pom.xml, fetched from Maven repositories. Plugins like content-package-maven-plugin automate deployment.

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

mvn clean install removes old builds (clean), compiles code, and installs the package into the local Maven repository.

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

mvn clean install -PautoInstallPackage

This builds and deploys the package to AEM

**15.Explain the purpose of different Maven profiles in AEM**

**(autoInstallPackage, autoInstallBundle).**

**autoInstallPackage:** Deploys the full package.

**autoInstallBundle:** Deploys only the OSGi bundle.

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

The purpose is that , dumplibs helps debug client libraries by listing all clientlibs, dependencies, and inclusion paths.

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

We can view client libraries using dumplibs by -/libs/granite/ui/content/dumplibs.html

It shows the client libraries loaded on the page.

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

Client libraries are structured in AEM as /apps/project-name/clientlibs/.