OSGI service:

//For creating osgi service we create one interface ,one model class,one service class.//

Why we need model class here ?

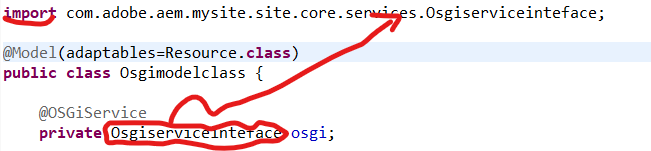
To call backend code to frontend we use the model class.

Why we need Interface here ?

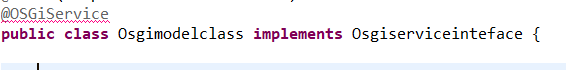
we need to call the osgi services to model class for that we store the inteface in the model class insted of implementing the interface , if we implement in the model class we can not insert the osgi services into the interface .

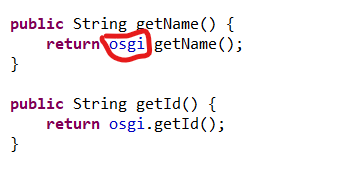


If you want to import osgi service directly insted of taking the interface but one problem occur , here actually we can not import the class but we can import the interface



If we implement the interface in model class we can not insert the osgi services,outside the method java not allowed to use the @OSGiService annotation,one more problem is there we can not use the reference name in the return type this is a another problem





Why we need Service class here ?

for creating the osgi component we need the osgi service class , it creates osgi service in system console we can reuse this service wherever we want

@Component: This is an annotation used in OSGi environments to declare a class as an OSGi component.

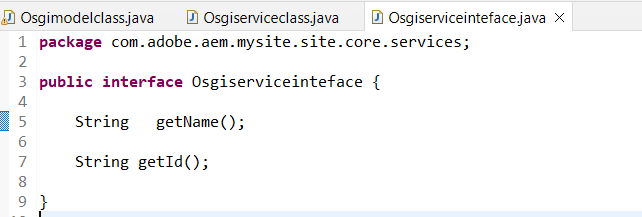
We can create OSGI service with the help of **@Component** annotation from org.osgi.service.component.annotations package and register it as an OSGI service.

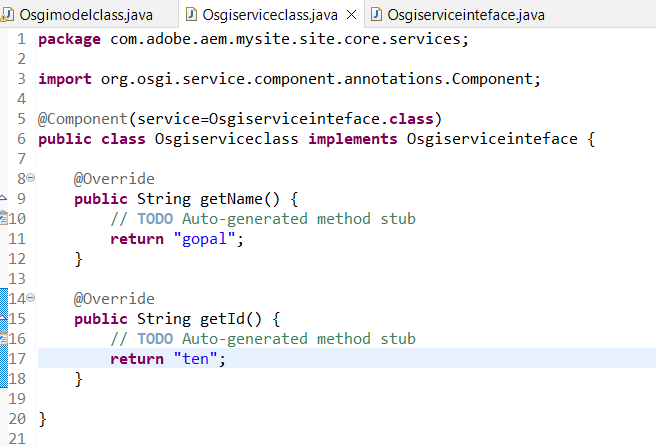
Services can be consume using **@Reference** annotation in Servlet and other services. User **@OSGiService** annotation n to inject service in sling model.

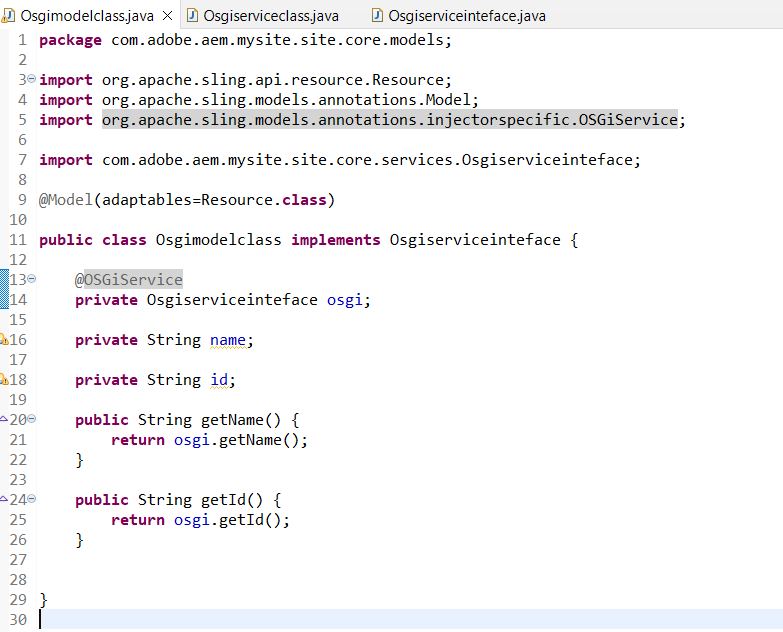
Certainly! In OSGi, when a component is defined with the immediate = true attribute within the @Component annotation, it means that this specific component will be activated and instantiated as soon as the OSGi framework starts, disregarding the default behavior of waiting for dependencies to be fulfilled.

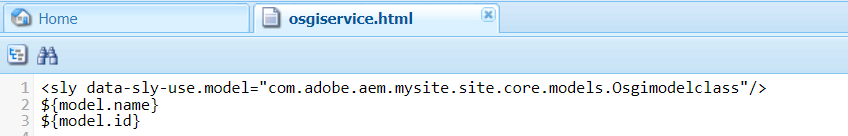
By setting immediate = true, the component doesn’t wait for its required dependencies to become available before being activated. Instead, it's immediately instantiated once the OSGi framework initializes, assuming all its dependencies are resolved during this initialization phase.

This attribute is particularly useful for components that are critical or standalone, meaning they don't rely heavily on other services or components to function properly. It ensures that such components are ready and available right from the start of the application, potentially reducing any delay or waiting time for their activation until all dependencies are fulfilled.









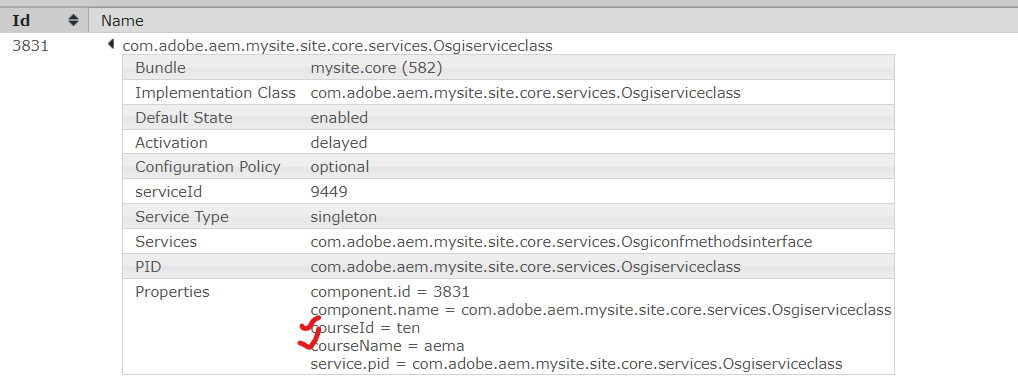
OSGI CONFIGURATION:

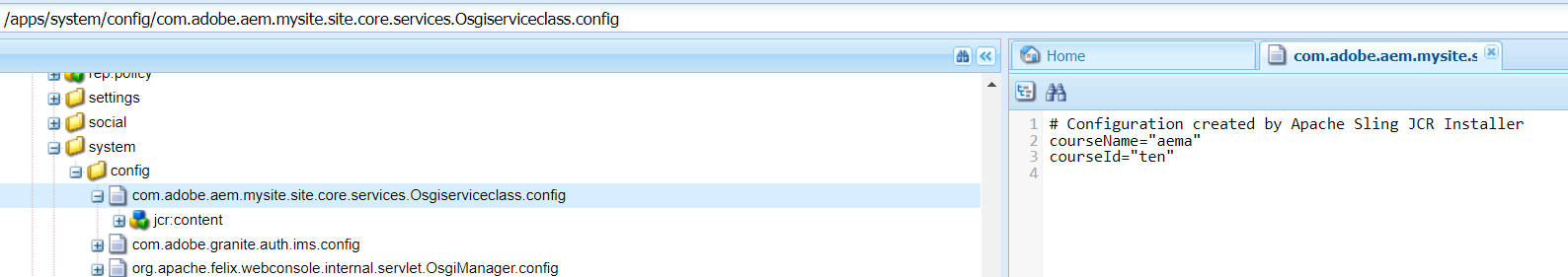
What is the use of osgi configuration rather than osgi service ?

Osgi configurations creates the bundle here we can give the input ,for suppose if we give the data in osgi service we have to run the java code to render the output on page ( by the way it not create the bundle ) but in osgi configuration we can set the data in bundle it will render the data on page without running the java code,this is the use of osgi configuration.

Why we need to give the @Designate annotation in osgiservice class rather than model class or other classes ?

If we give in osgiserviceclass it will add the data in osgicomponent because of this we can give the input to the bundle in crxde directly





@Designate:add configurations to OSGI Services with the help of @Designate annotation. @Designate is must to have ocd(object class definition) property to declare service name as mentioned in below syntax.

@Designate(ocd = PracticeServiceOCDConfiguration.class)

@interface :Generally in normal Interface we will give the return type(data) in java itself only but in the osgi configuration we give the data in the osgi bundle ,it is outside of the data that is why we are using @Interface

@PostConstruct: This annotation can be used to run the logic once all the field level injections are done.

