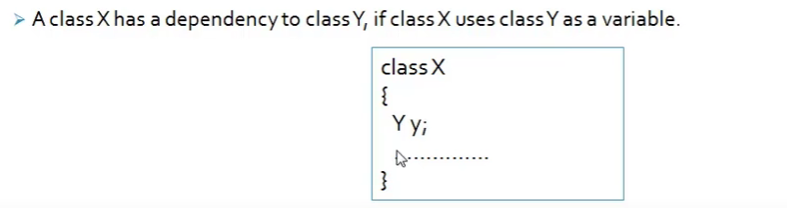
WAR file (Web Archive)

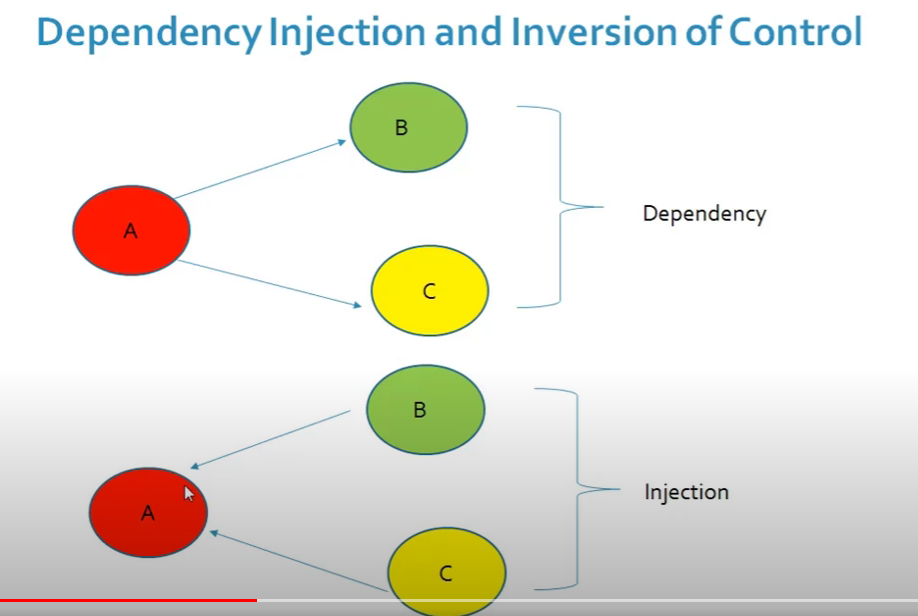
Spring Boot Framework

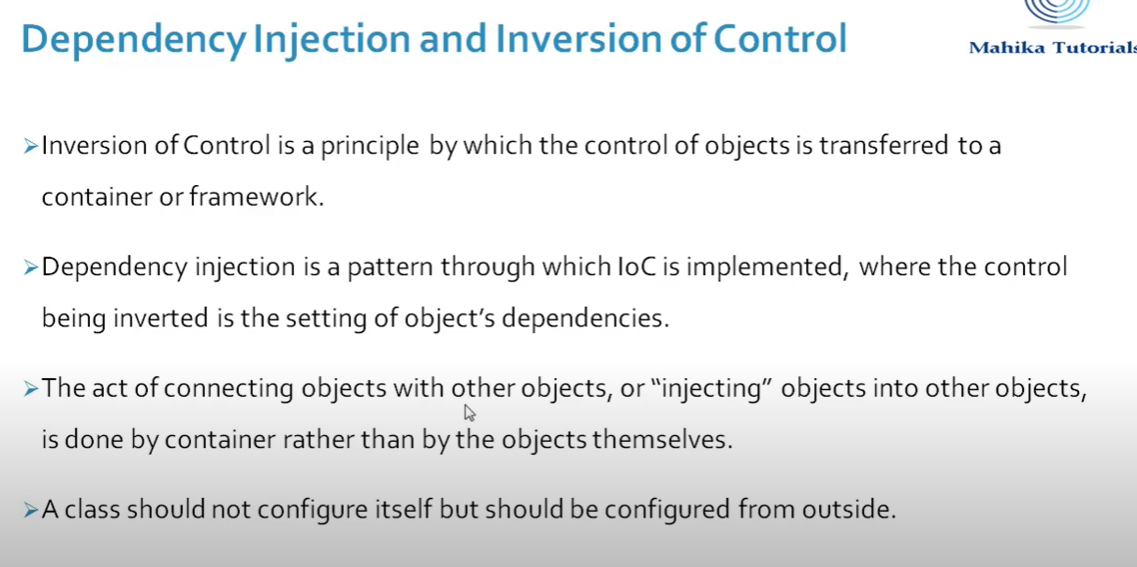
Dependecies Injection

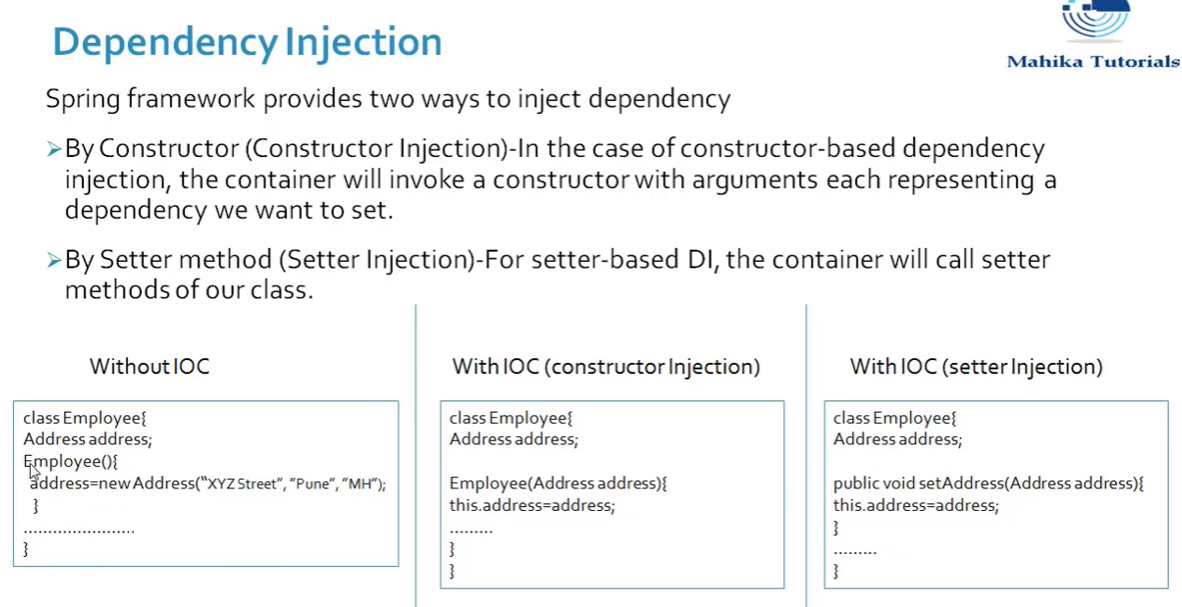
**1 Inversion of Control**

Giving control to frame work to do that spring allows dependency injection.









**2 AOP(Aspect oriented programming)**

Tackle down the cross cutting applications (repeatedly work)

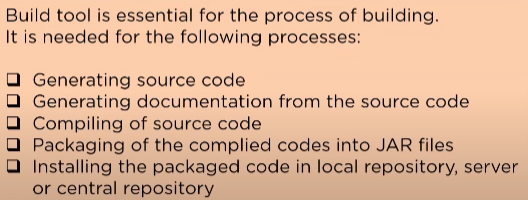
**3 Web Application**

Spring data libraries to connect with database easily (not tedious jdbc)

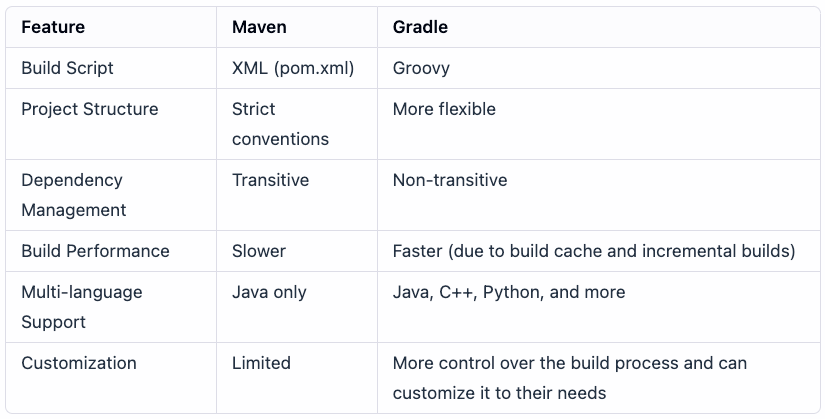
Ease of implementing other frameworks so its called **frameworks of frameworks**

**Spring Bot is a layer on top of the java to build our applications**

Maven is build tool helps in project management.



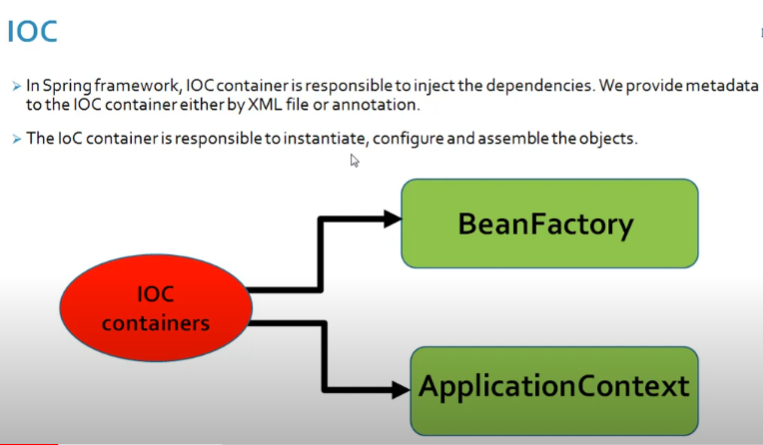
Maven mainly used for java-based projects, helps in downloading dependencieds, which refers to the libraries or **JAR files.**

****

* **groupId – a unique base name of the company or group that created the project (unique)**
* **artifactId – a unique name of the project**

**Dependency Injection**

the objects that form the backbone of your application and that are managed by the Spring IoC container are called beans



When ever spring starts we get beans from container

IOC(Inversion of control) has 2 interface **Beanfactory** and **ApplicationContext**

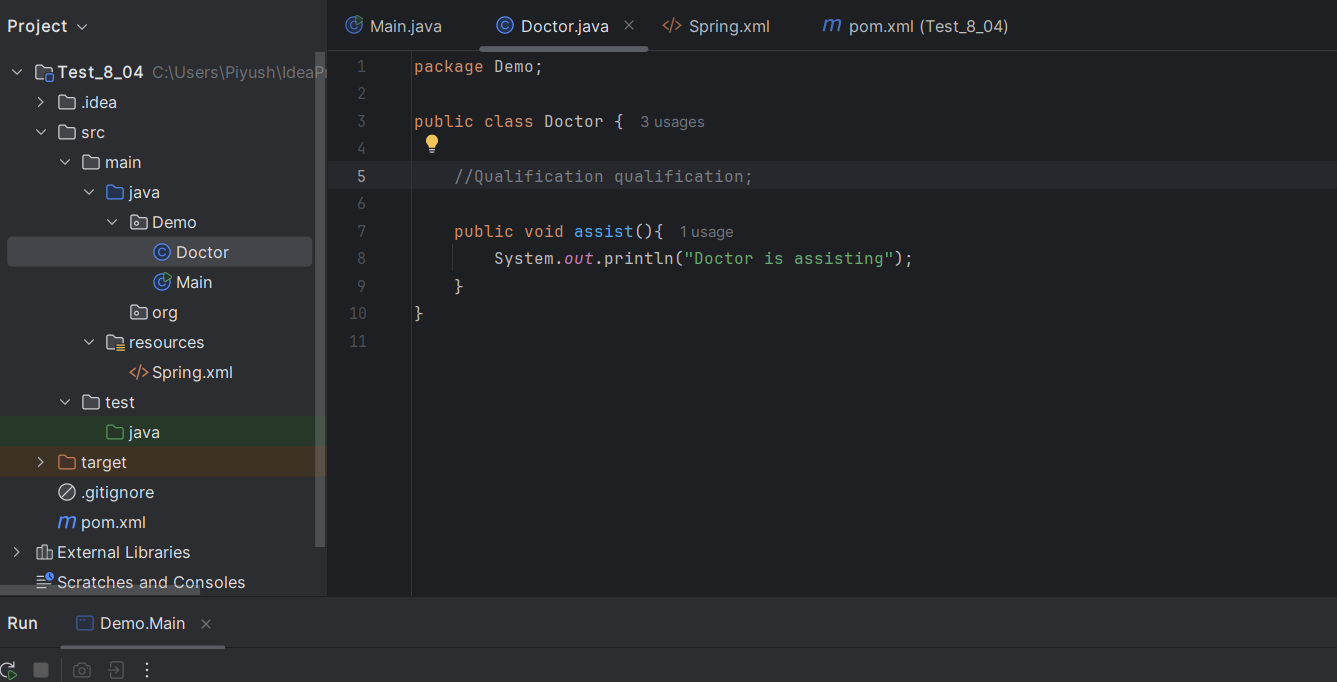
The BeanFactory provides the configuration framework and basic functionality, while the ApplicationContext adds enhanced capabilities to it.

clas

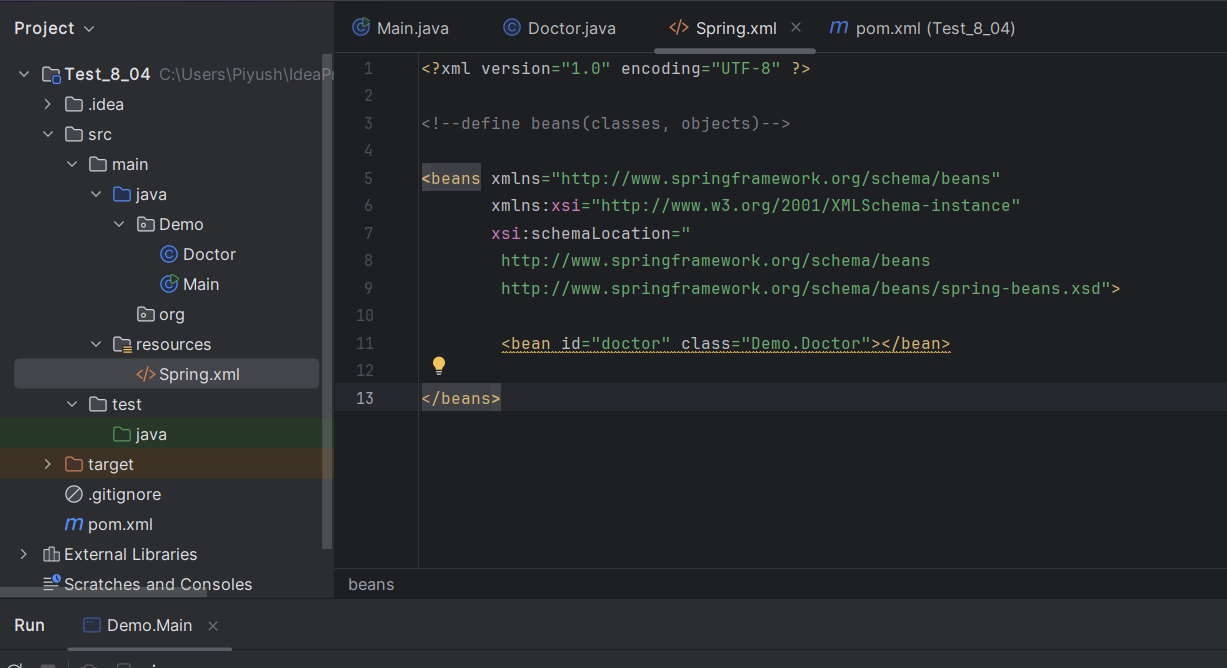
ApplicationContext extends beans factory . so there are lot more features in ApplicationContext

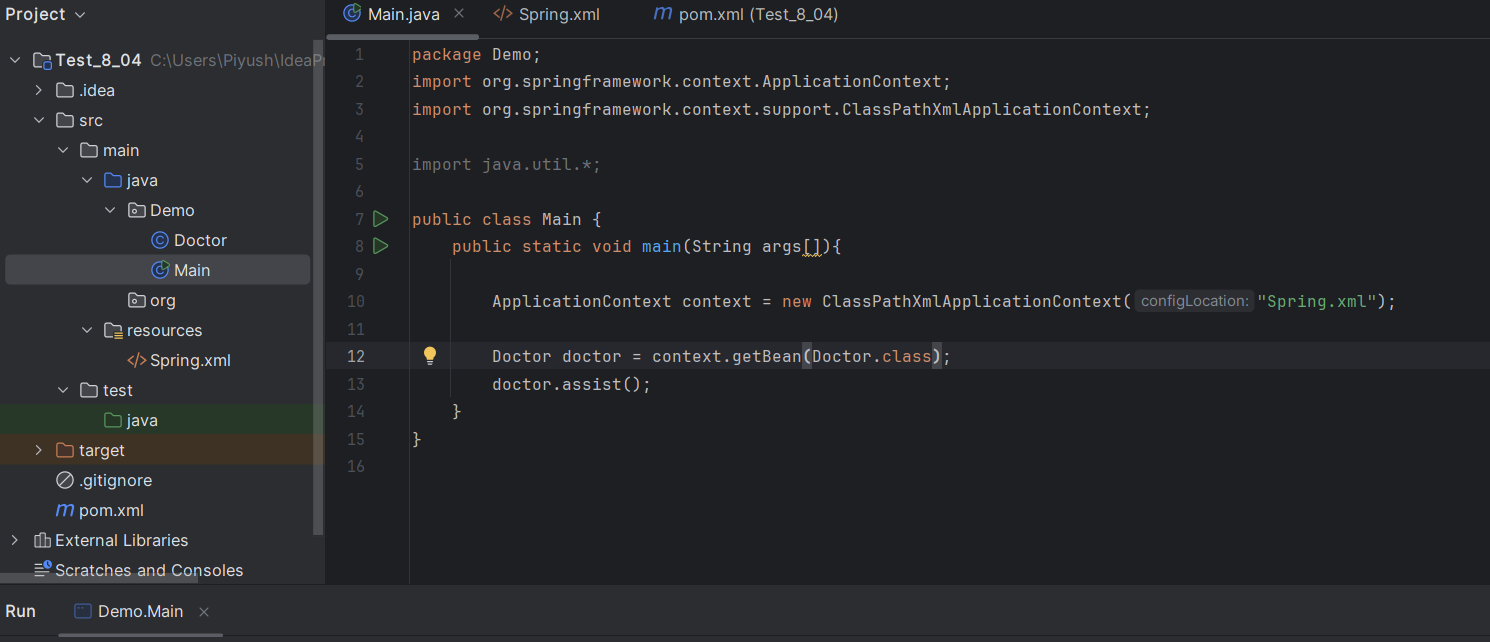
This try to get all beans

Class **ClassPathXMLApplicationContex**t implements the ApplicationContext interface

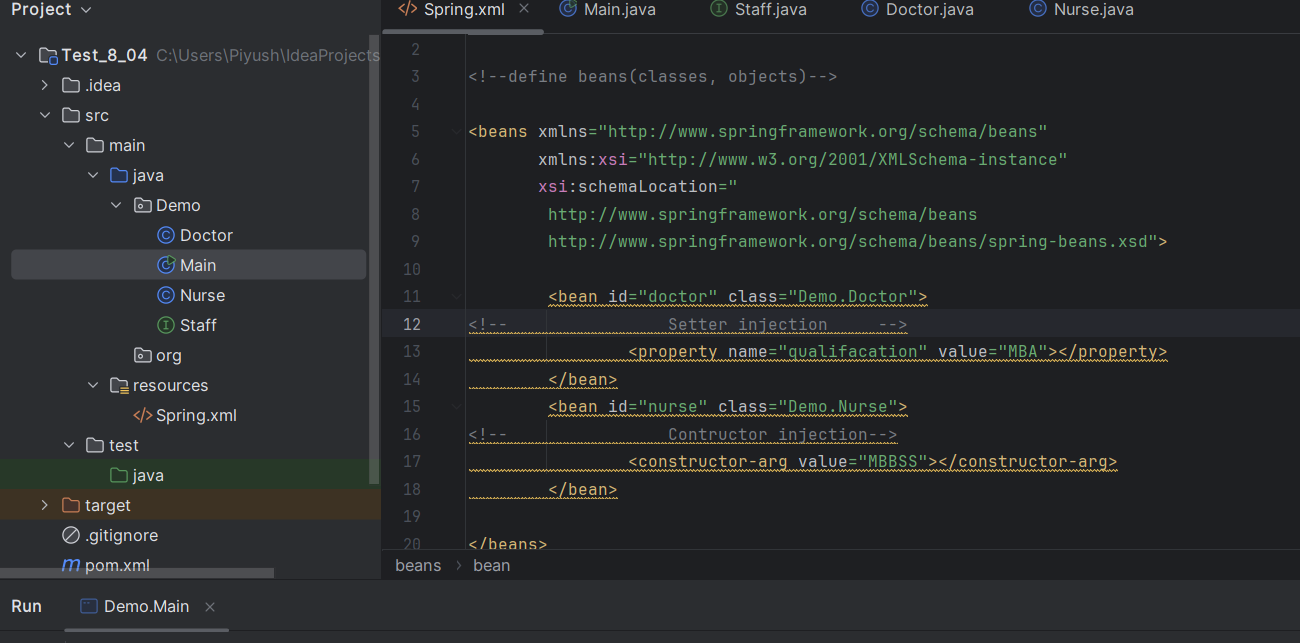


Defining xml file for Bean

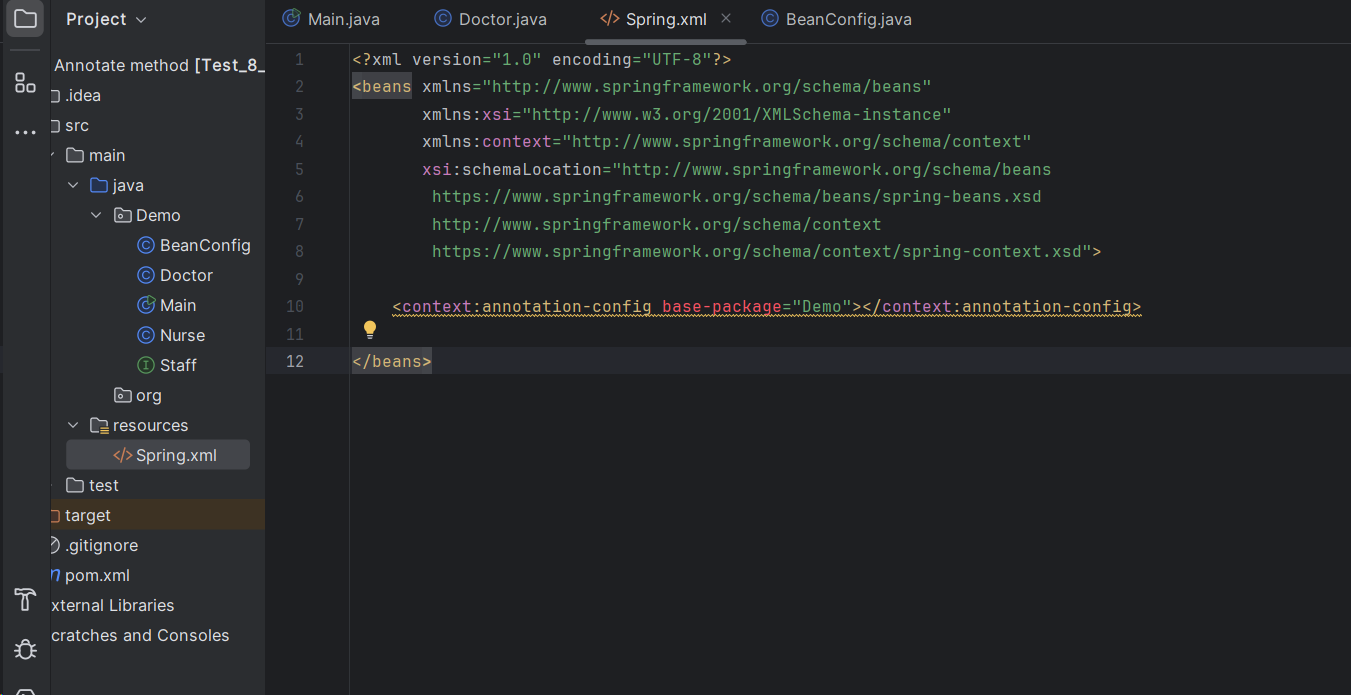


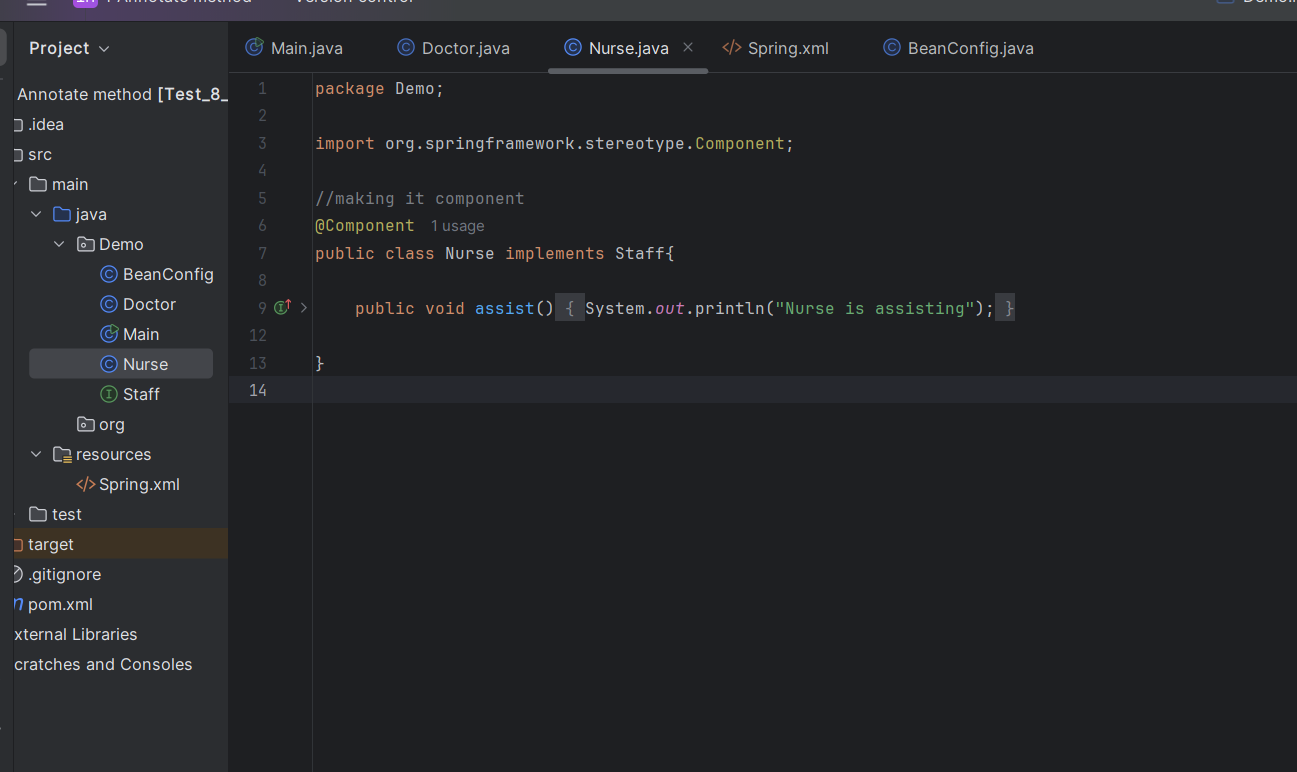


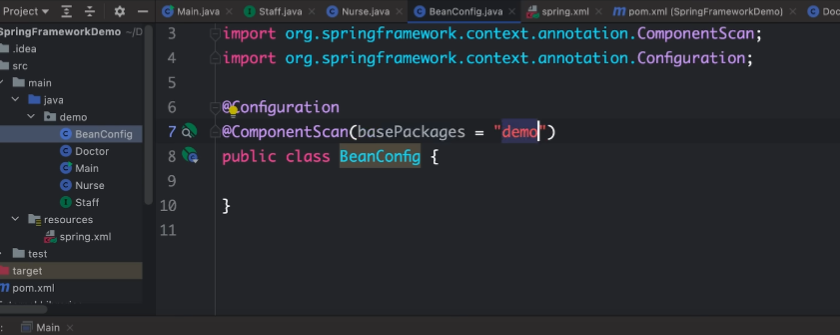
Setter Injection and Constructor Injection into beans

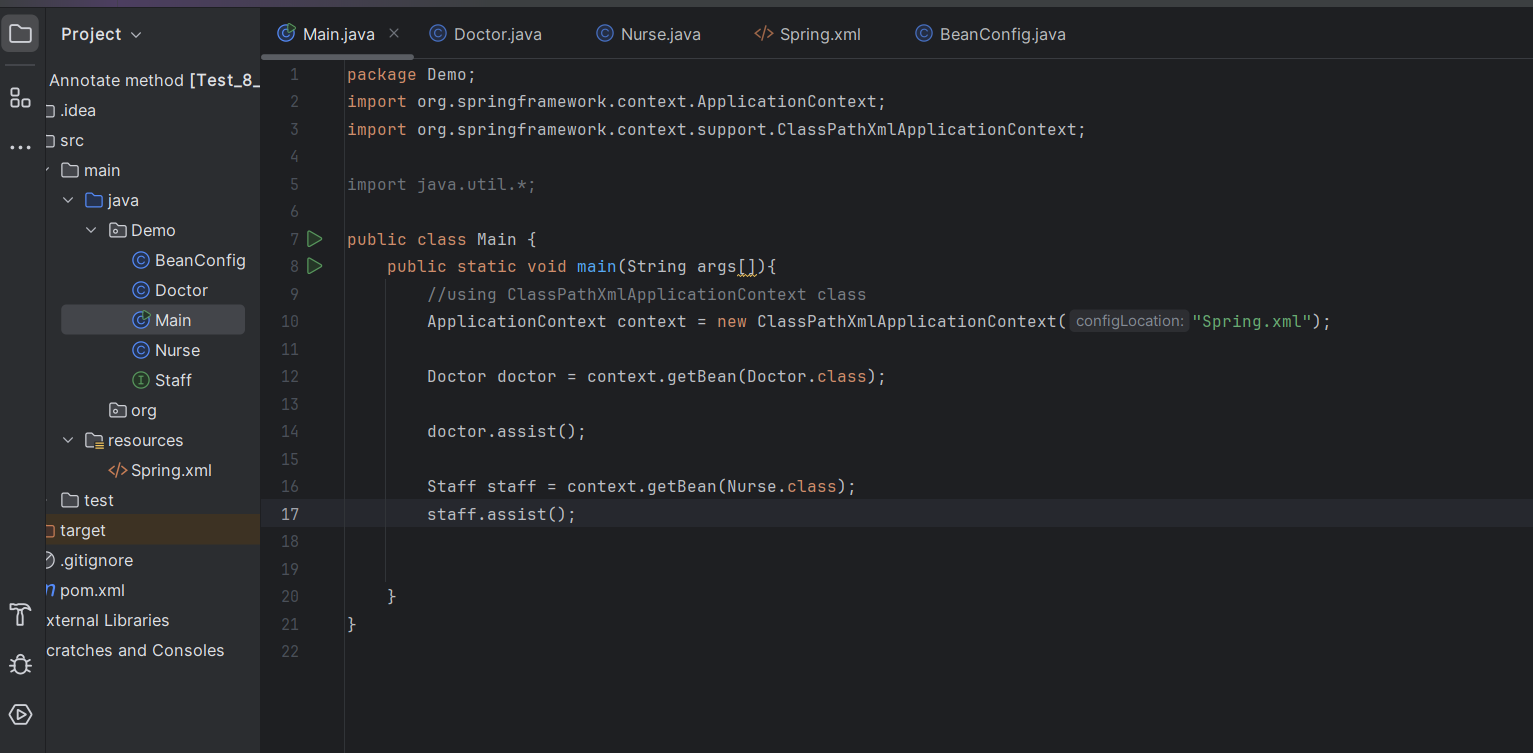


Using Component annotate





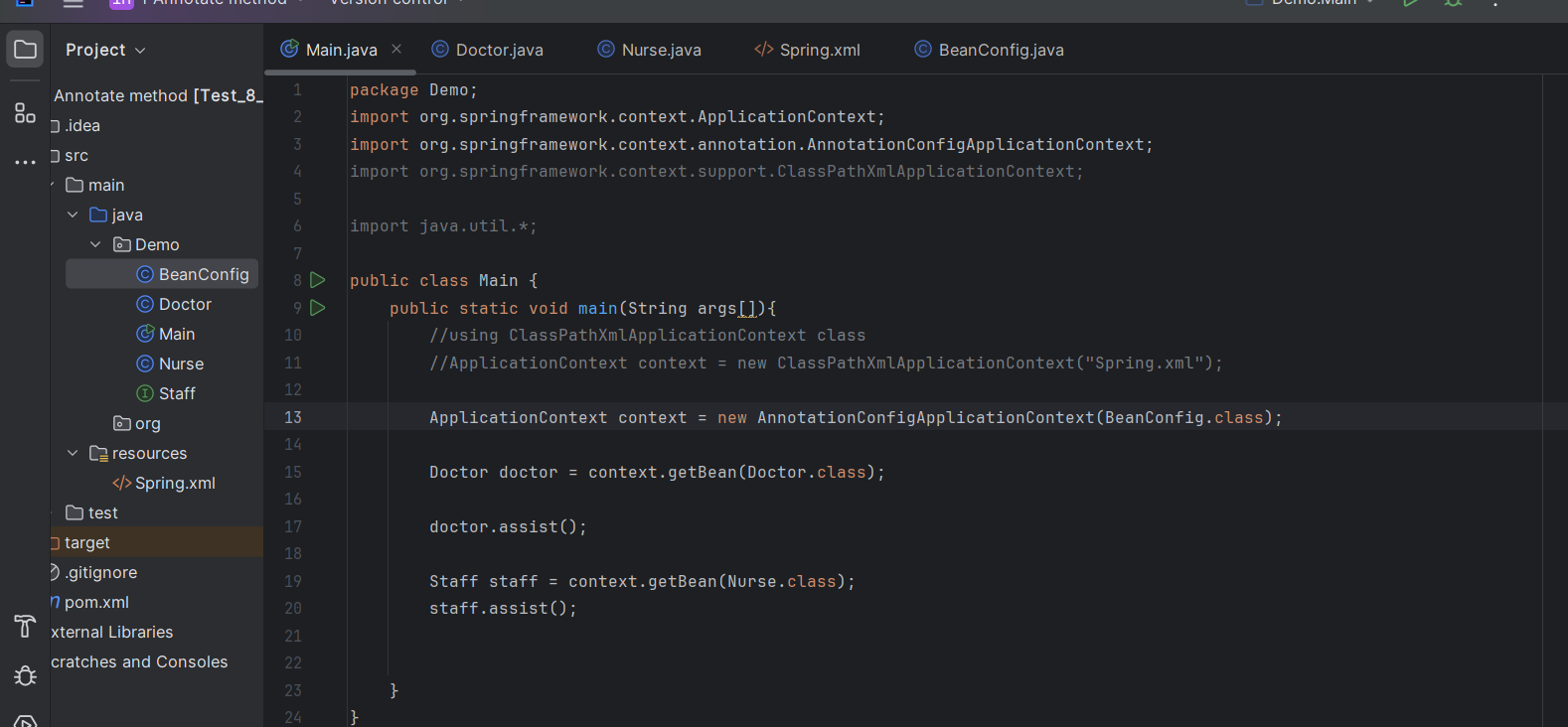




Here it will go to BeanConfig class, check what all package to be load where @Component is defined.

**Defining in bean Class**

****

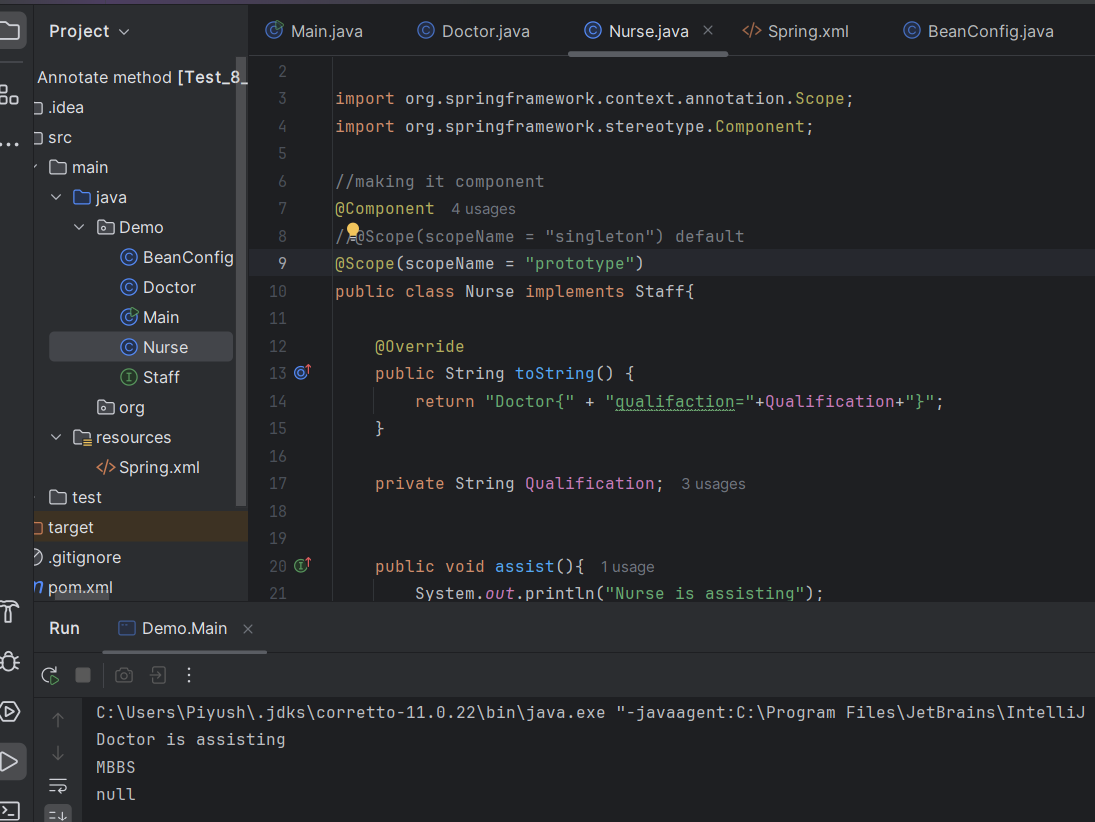
****

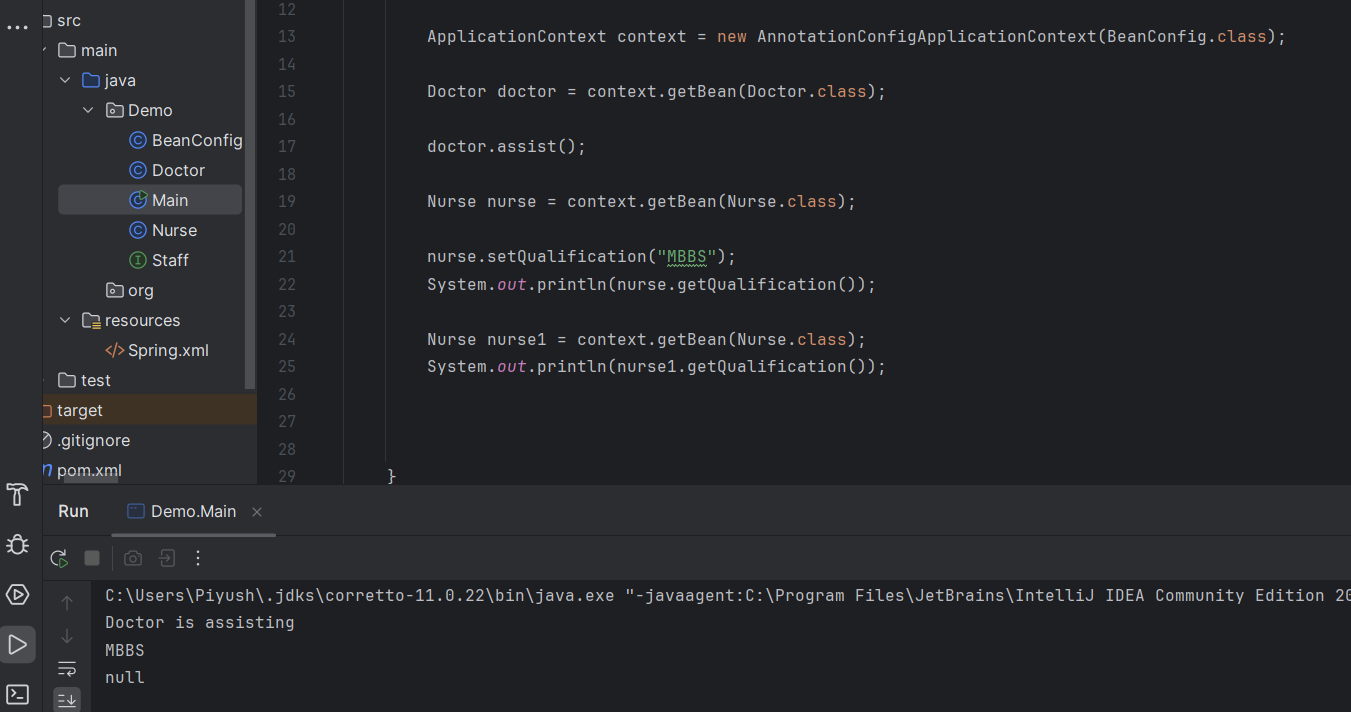
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Types of scope**

Singleton (default scope)(All access only one object)

Prototype (Different objects)





request

session

application

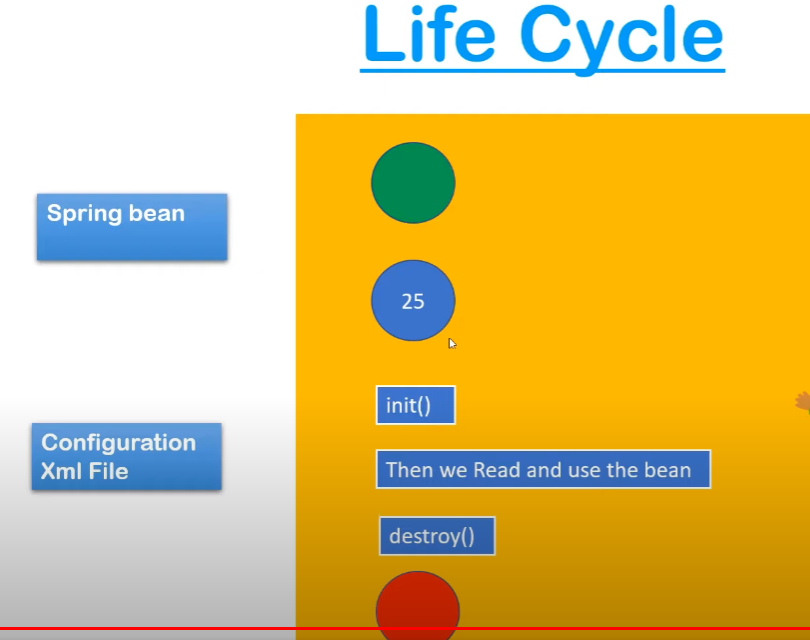
Websocket

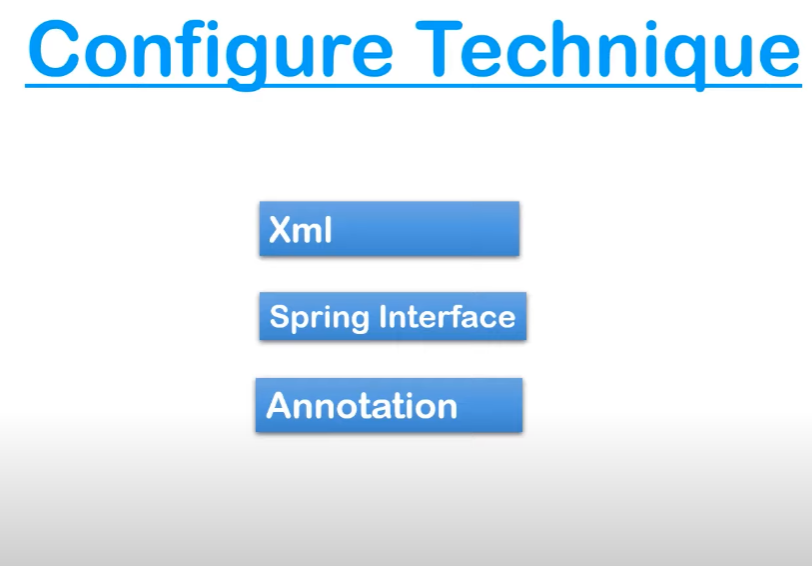
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

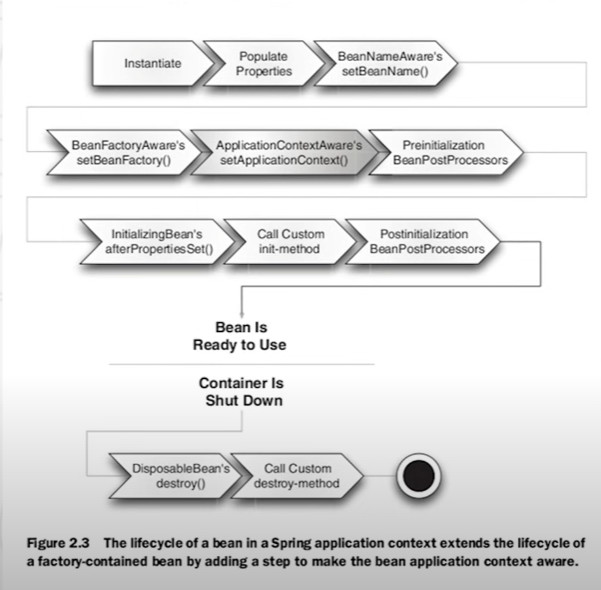
Bean Life Cycle



**Bean accepts xml file**

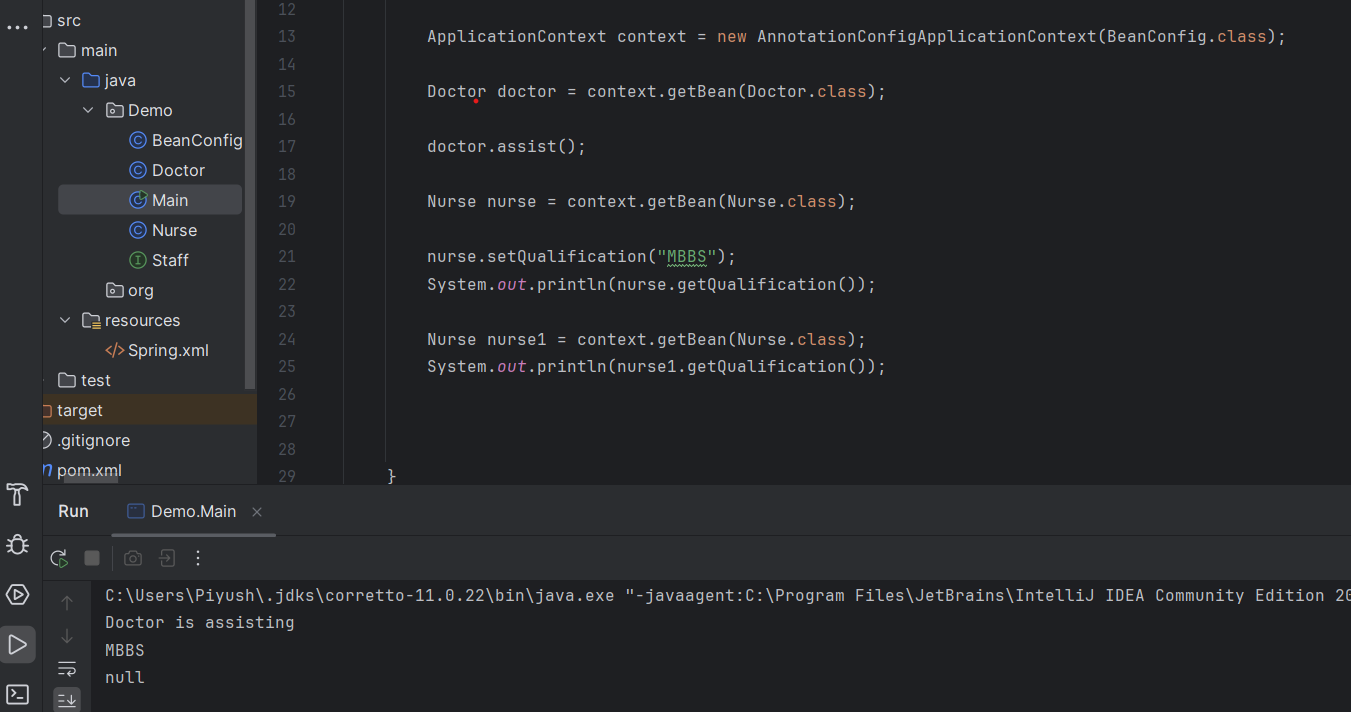
****

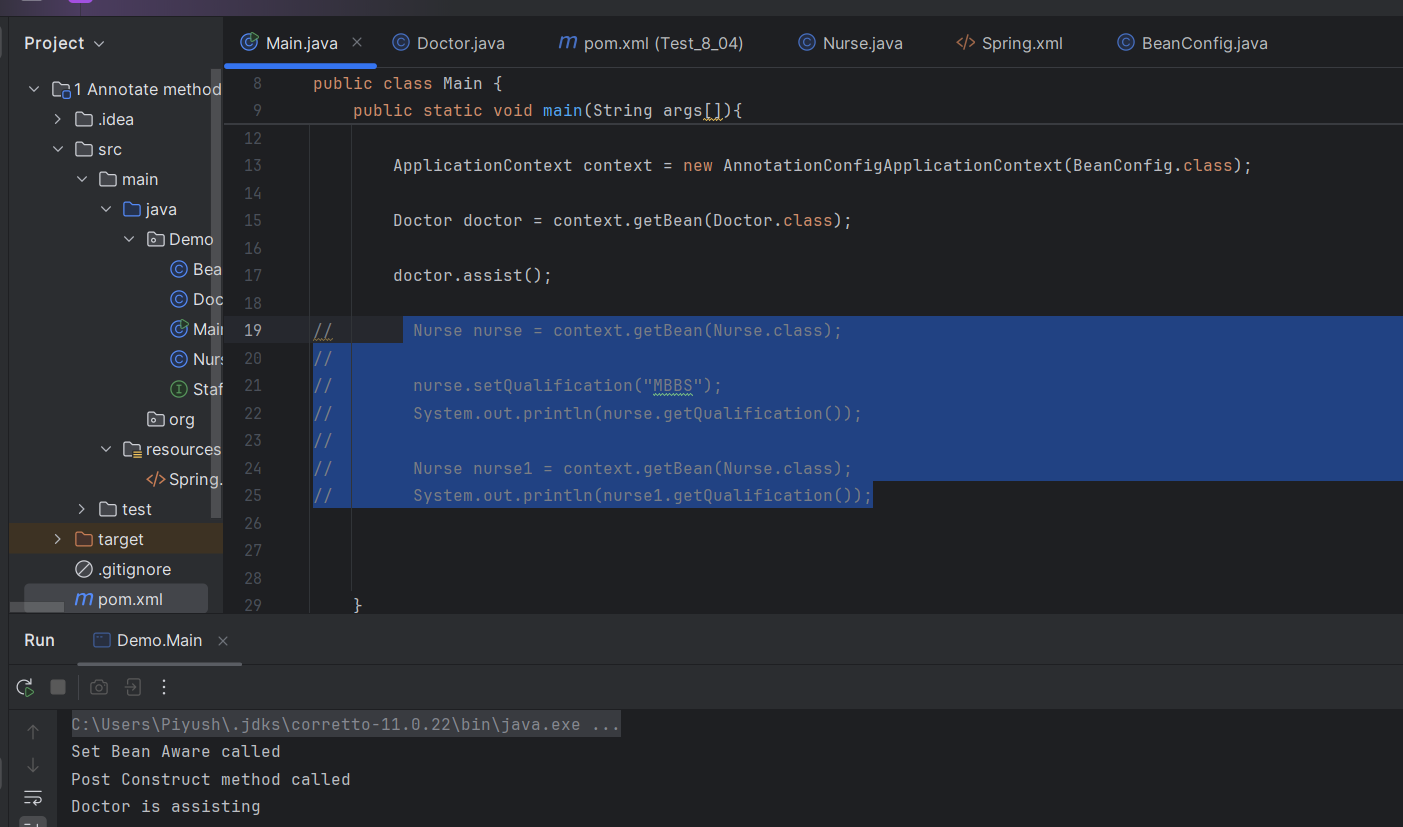
****

****

**SetBeanMethod**

**Post construct**

****

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

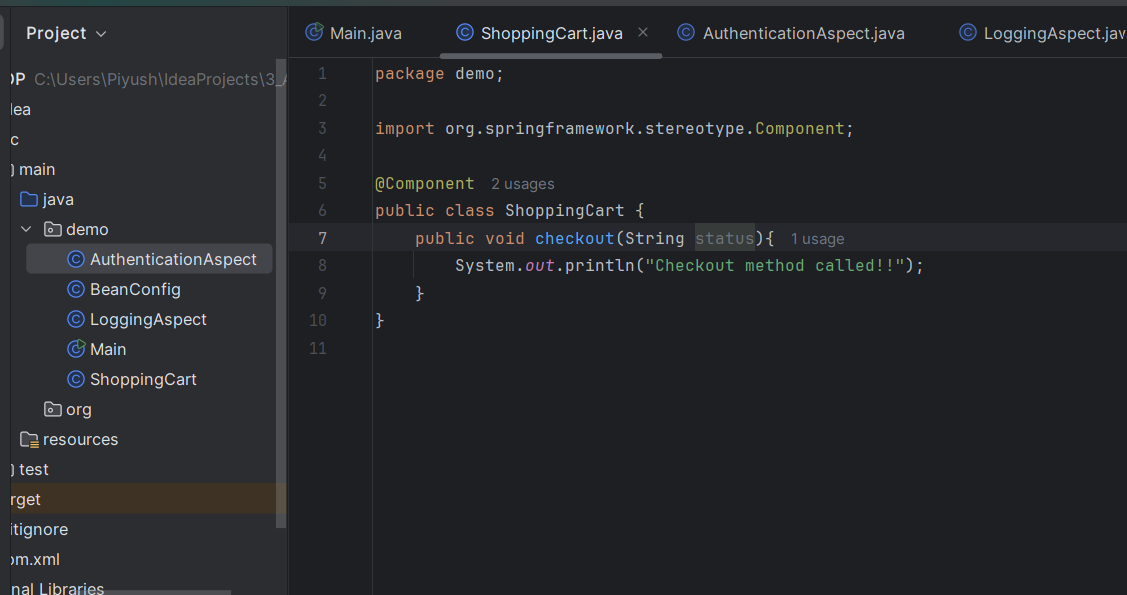
**Bean then no Components**

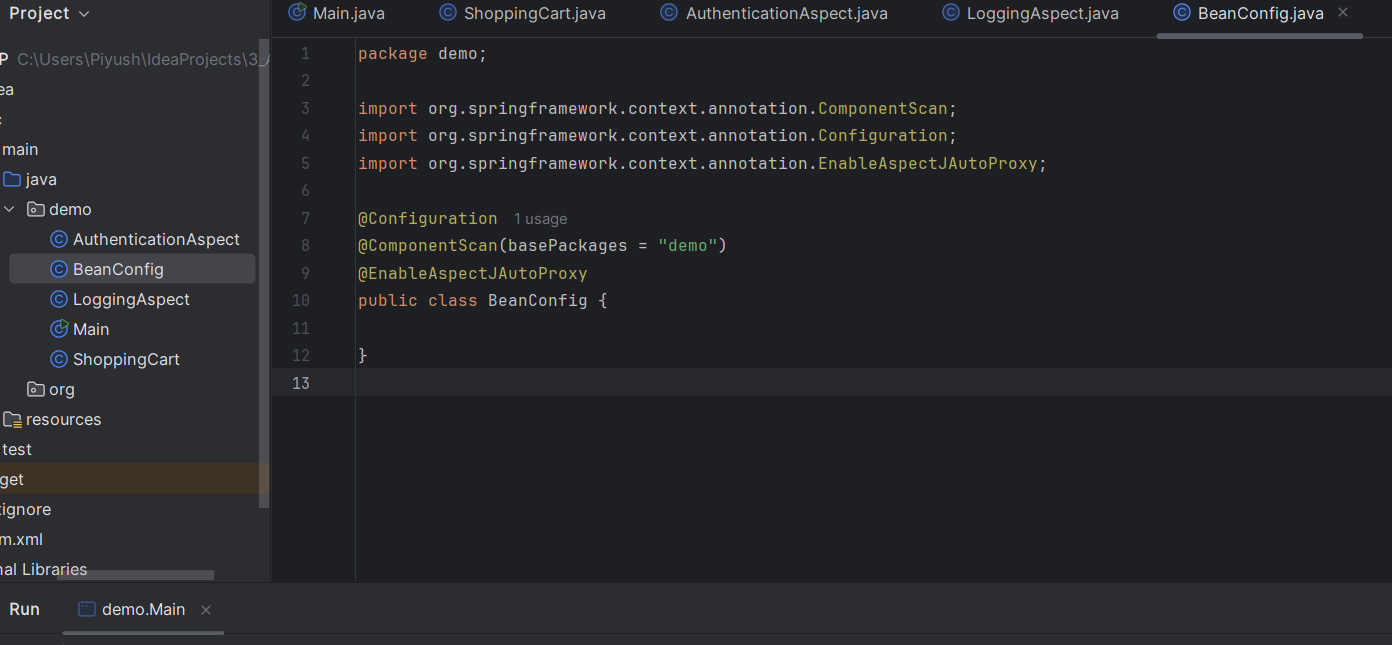
**If Components then no need to define Bean**

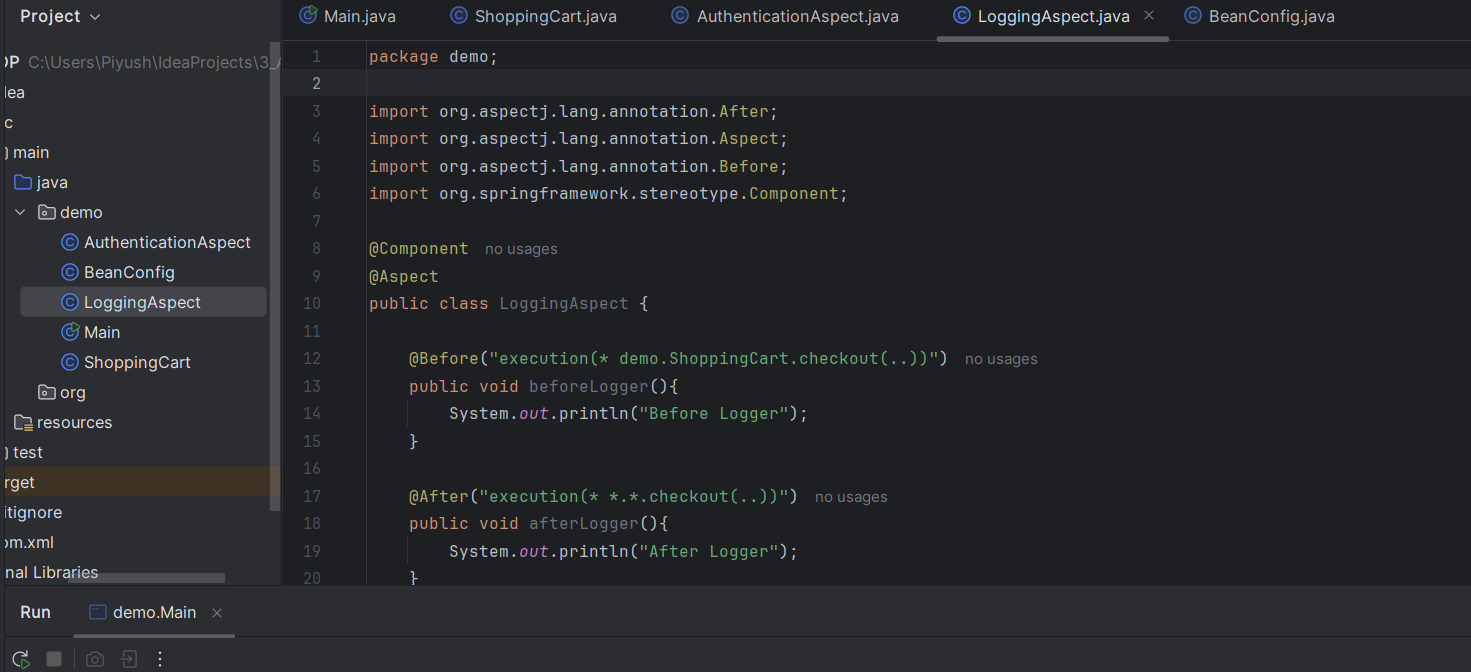
**AOP**

Remove cross cutting concern(loggers,authentication,sanitizing the data)

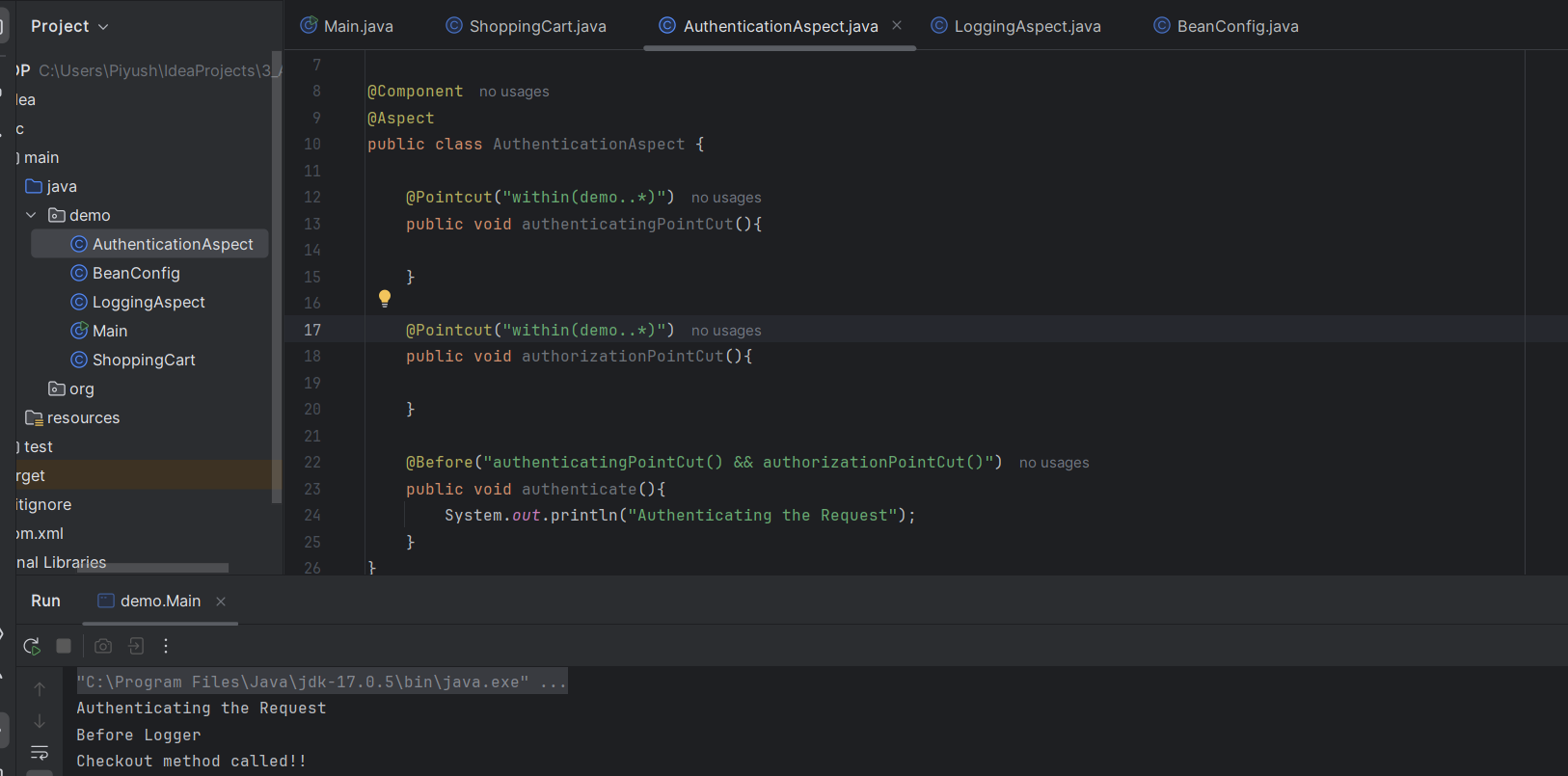
Loggers







Point cuts for authorization



**Spring Boot is extension of Spring framework (Rapid Application Development)**

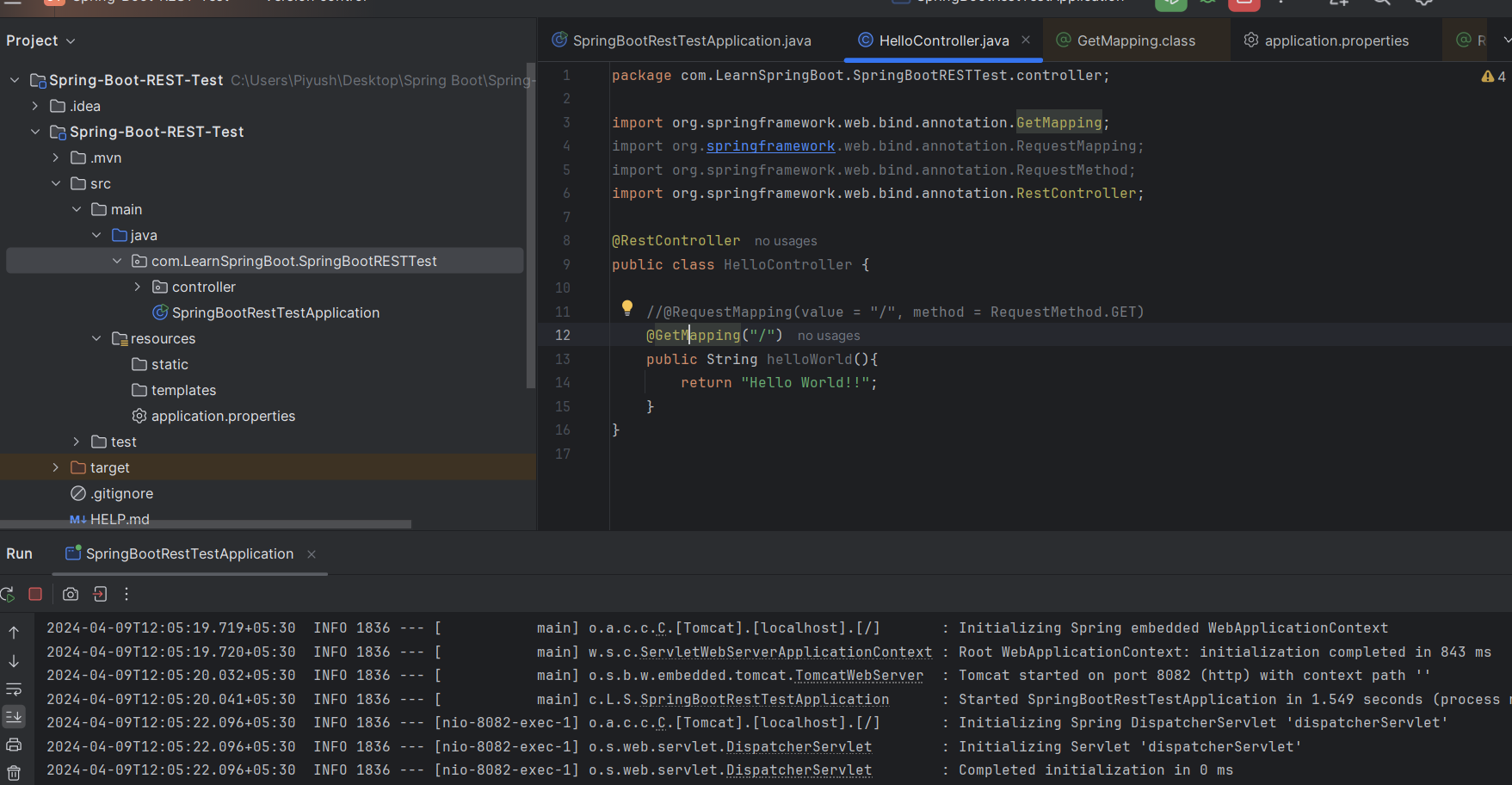
**Starter Template** For all(jdbc, test)

Auto Configuration

Embedded Servers (JAR File)(Always production ready)

Micro Services (smaller, independently deployable services)

Tomcat is an open-source web server and servlet. The Apache Software Foundation has developed it. It is used widely for hosting Java-based applications on the web.

****

****

**Controller Layer for (RestAPI’s)**

**Service Layer(Business Logics)**

**Repository Layer(Interaction with Database)(Reference to DB)(SprigDataJPA Dependency)**

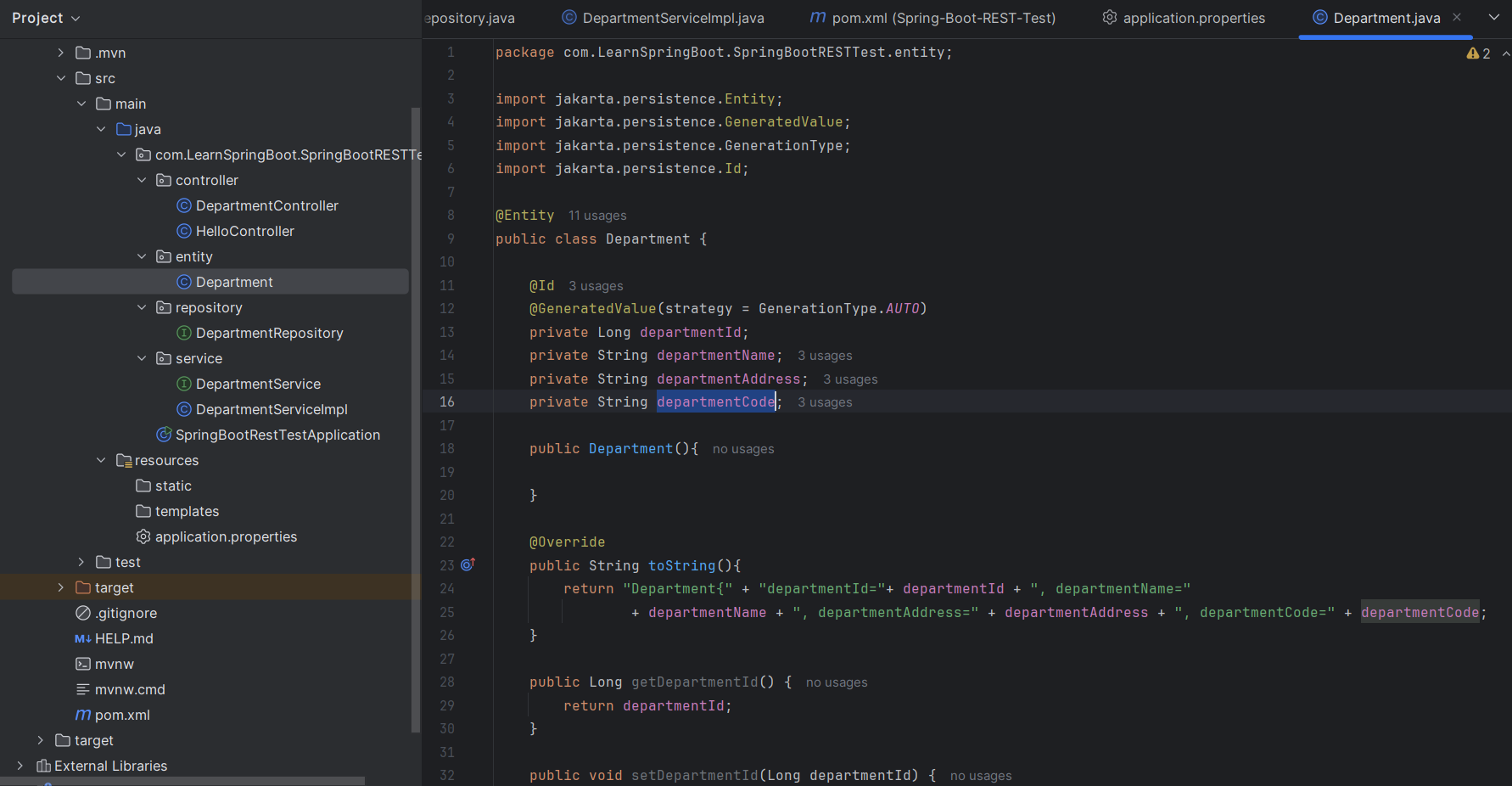
**DataBase(Instantiate DataBase Connection)(Database configuration should be added)**

**Spring Data JPA, part of the larger Spring Data family, makes it easy to easily implement JPA-based (Java Persistence API) repositories. It makes it easier to build Spring-powered applications that use data access technologies.**

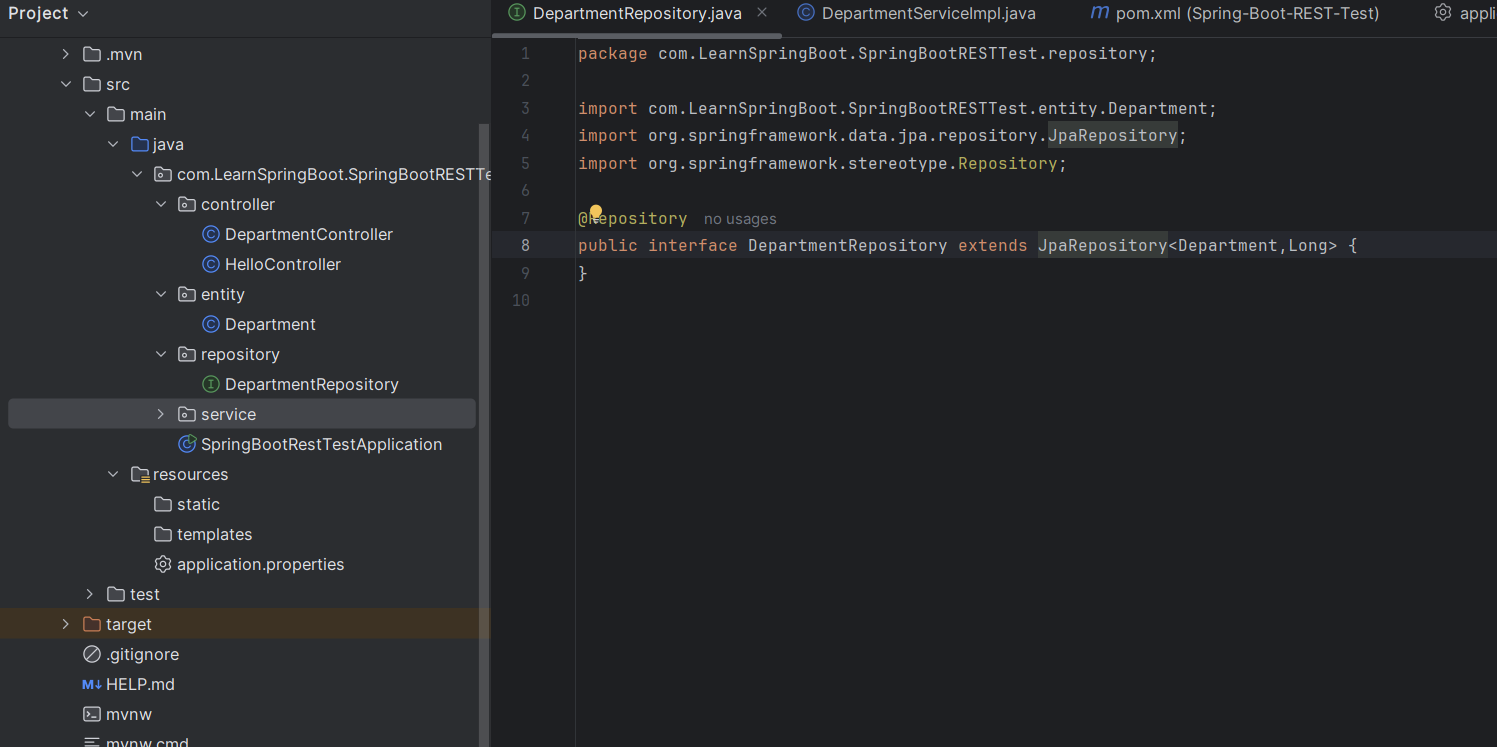
**Setting configuration.**

****

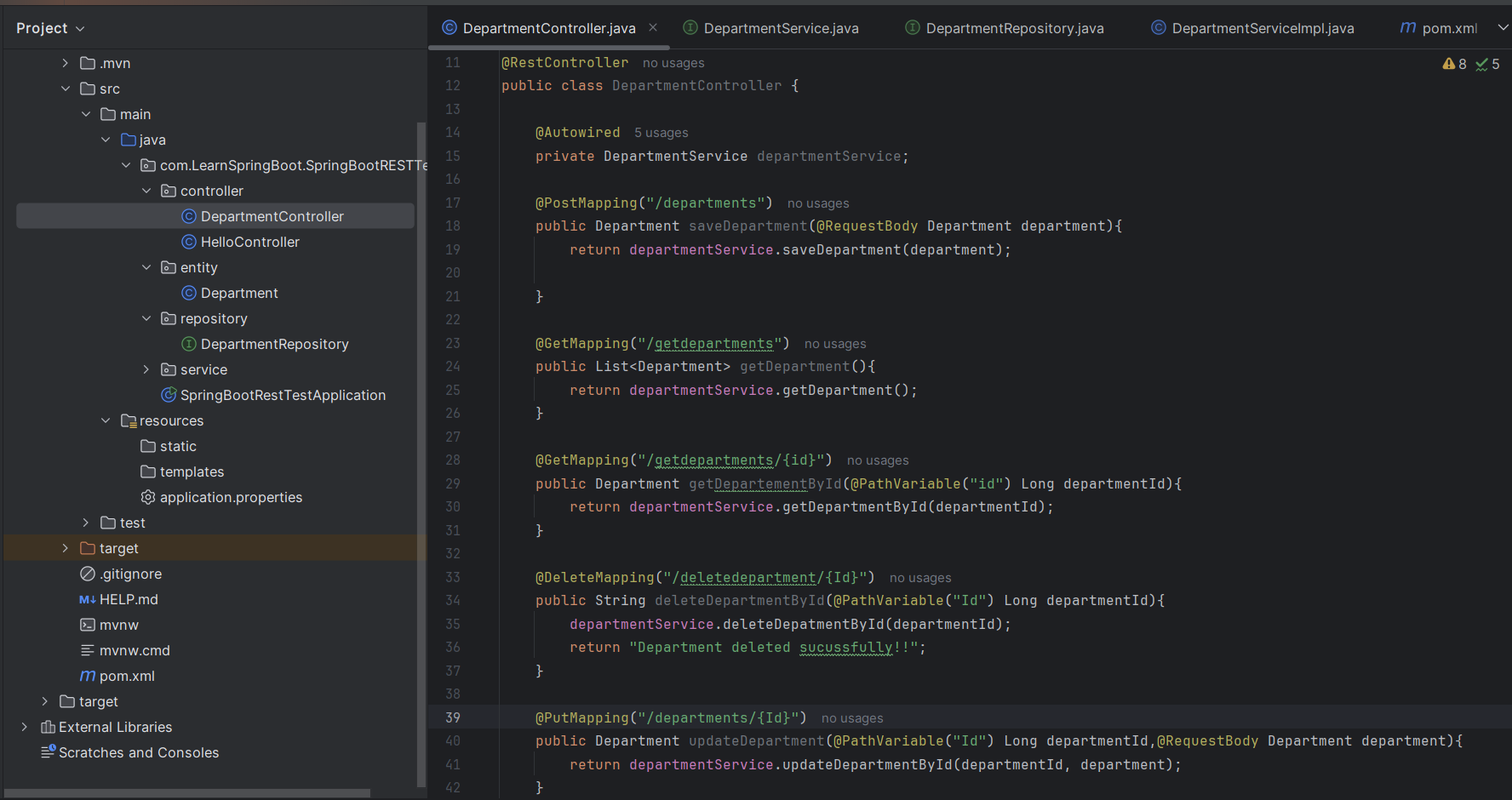
**1 entity model with getters and setters**

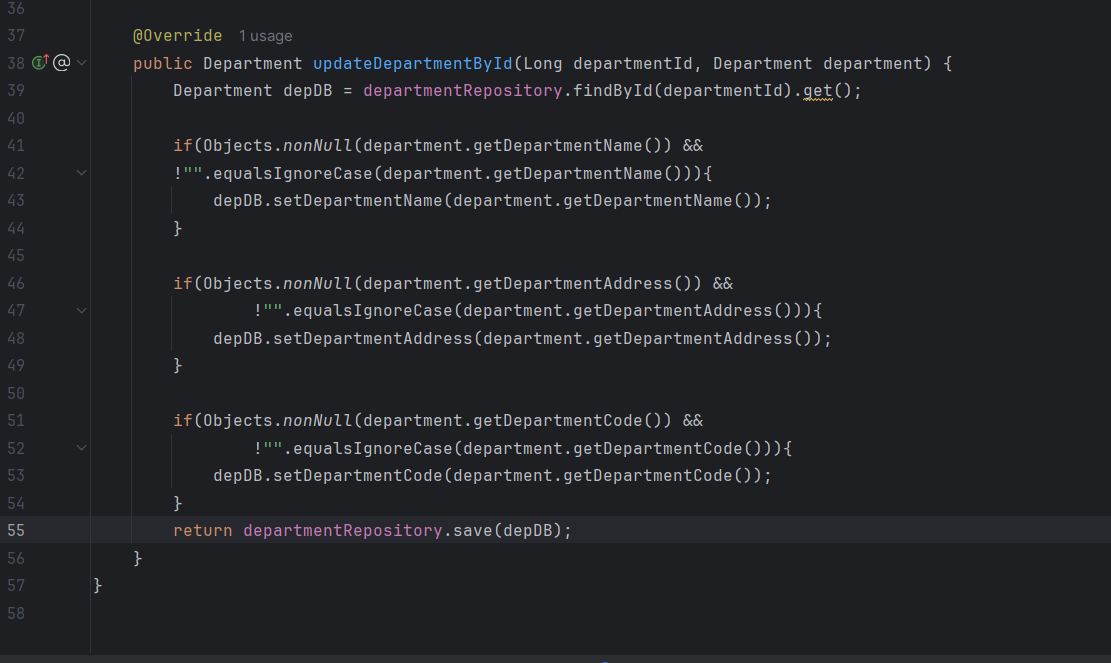
****

**2 Repository Layer(Interface) (JPA module extension)**

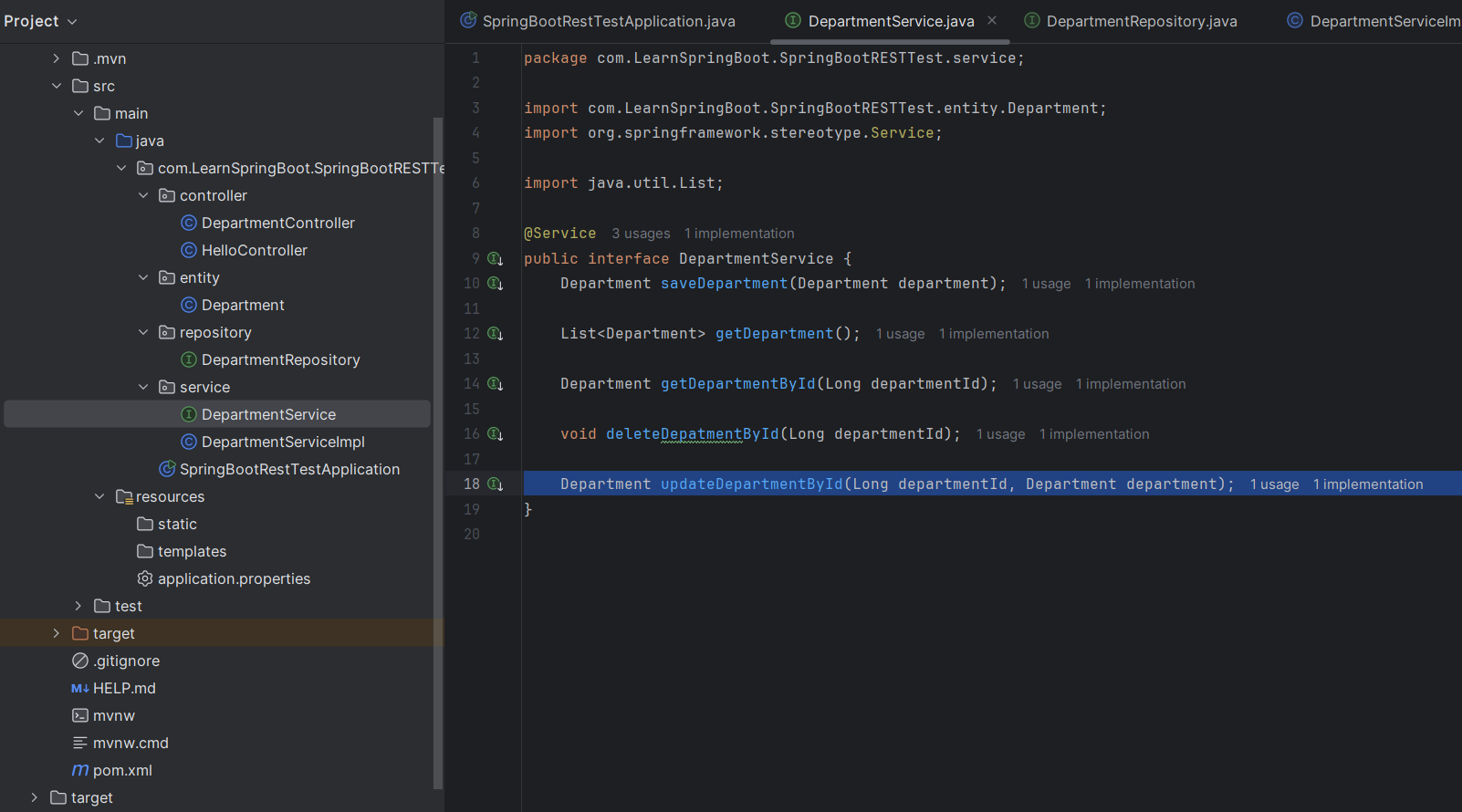
****

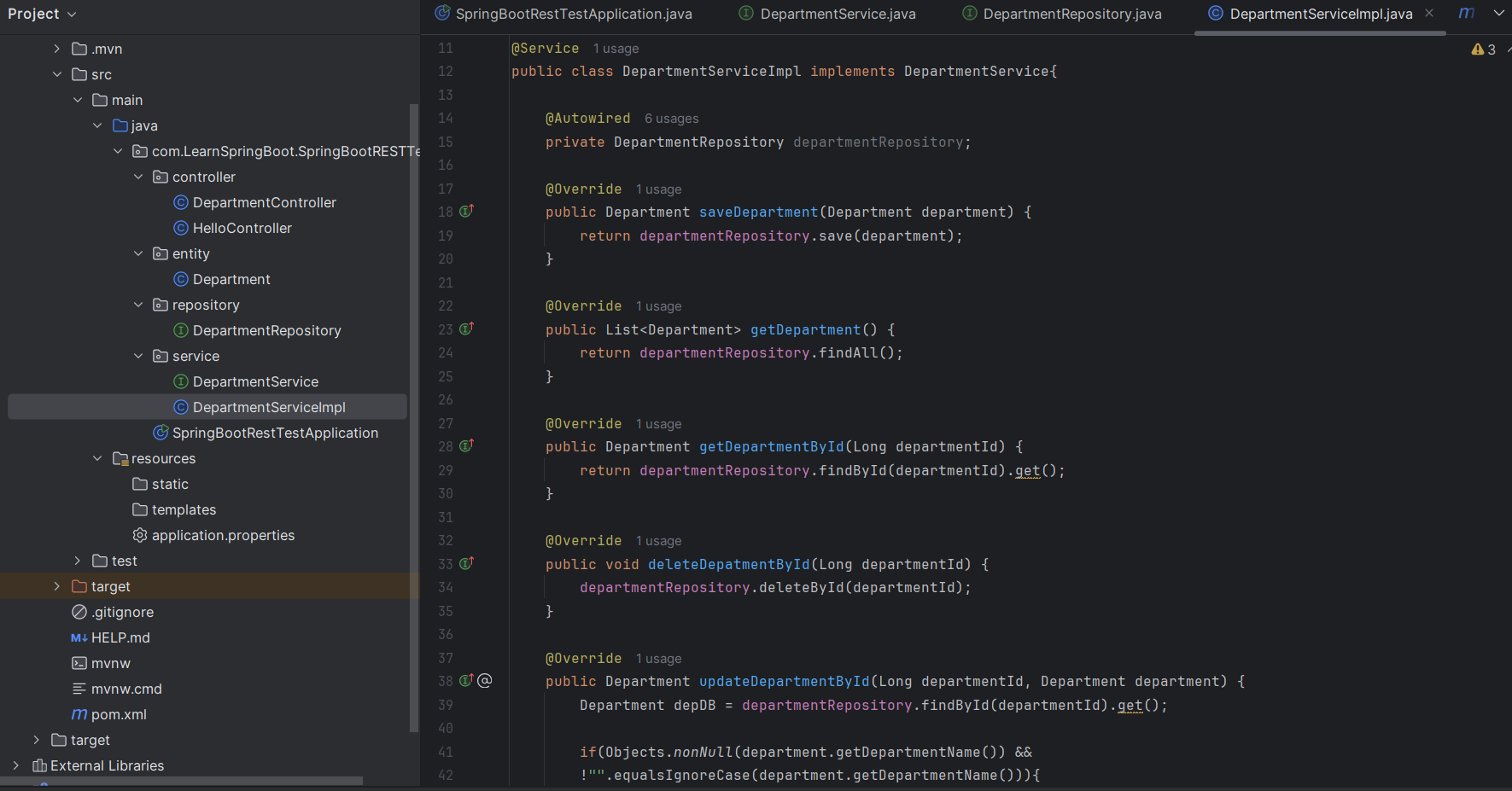
**3 Controller Layer (Define End points and service layers function calls)**

****

****

**4 Service Layer all the logic and implementation**

****

****