**SPRING CORE & MAVEN** **EXERCISES**

**( WEEK 3)**

**SUPER SET ID:6410152**

Exercise 1: Configuring a Basic Spring Application

**Step 1: Create a New Spring Boot Project**

You can do this using [Spring Initializr](https://start.spring.io/):

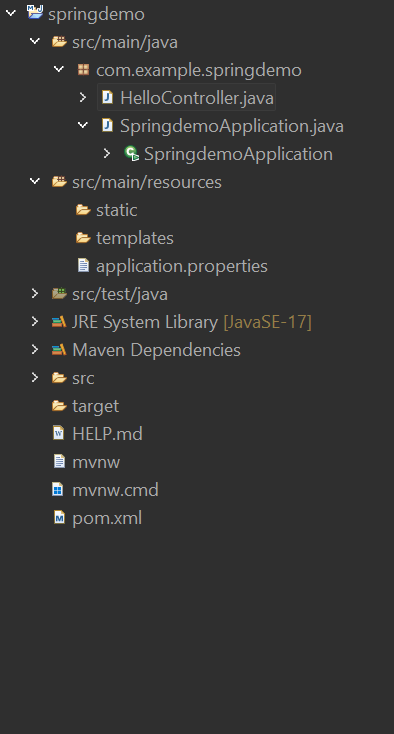
* **Project**: Maven (or Gradle)
* **Language**: Java
* **Spring Boot**: Choose the latest stable version
* **Project Metadata**:
  + Group: com.example
  + Artifact: spring-demo
  + Name: SpringDemo
* **Dependencies**: Spring Web

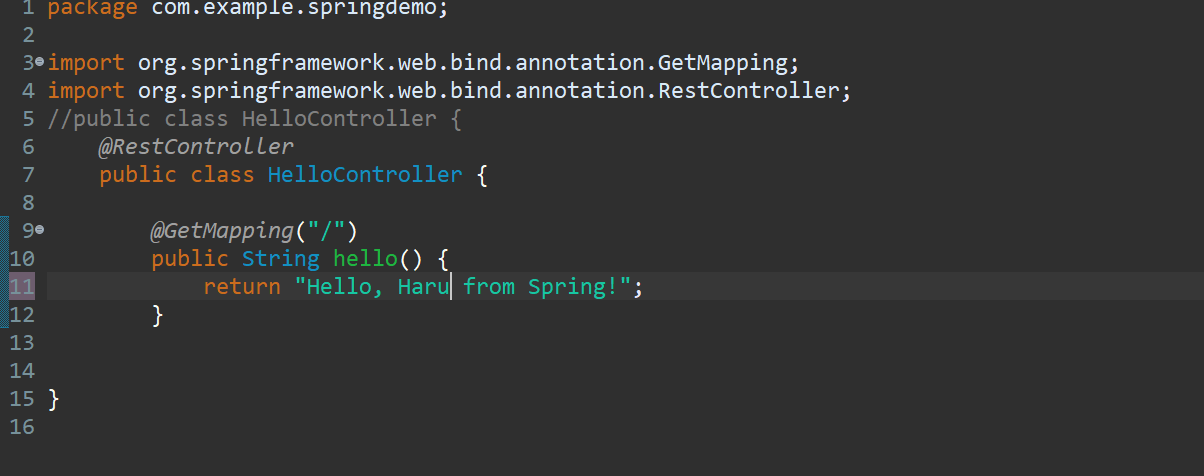
Click **Generate** to download the project zip.

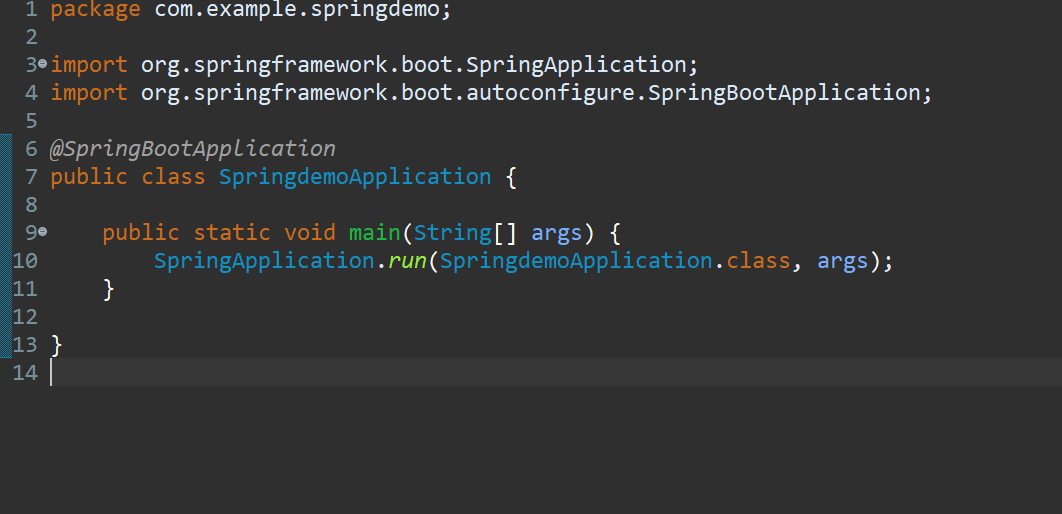
**Step 2: Import into IDE**

* Open Eclipse.
* Import the project as a **Maven** project.
* Wait for dependencies to resolve

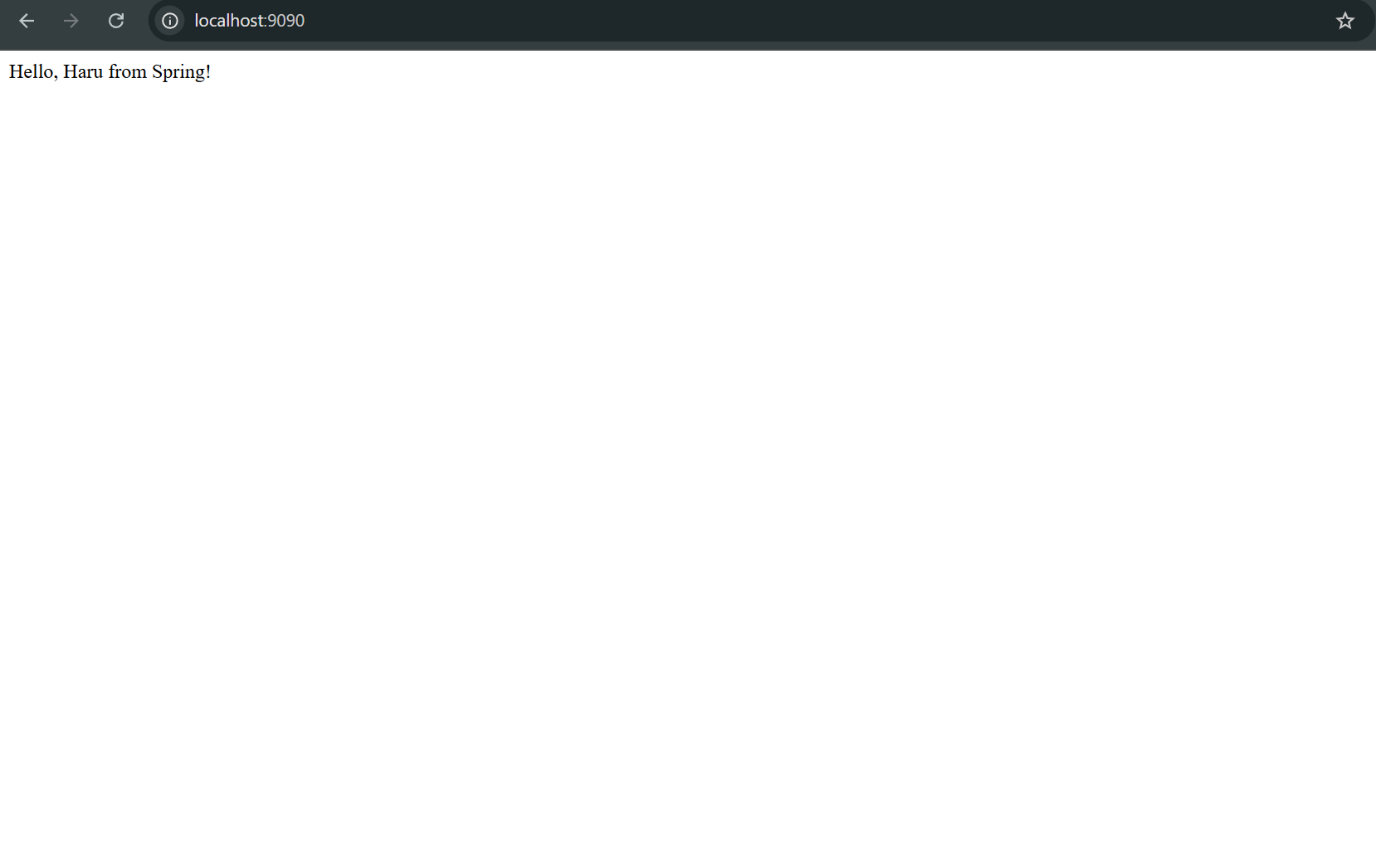
**Step 3: Structure**

****

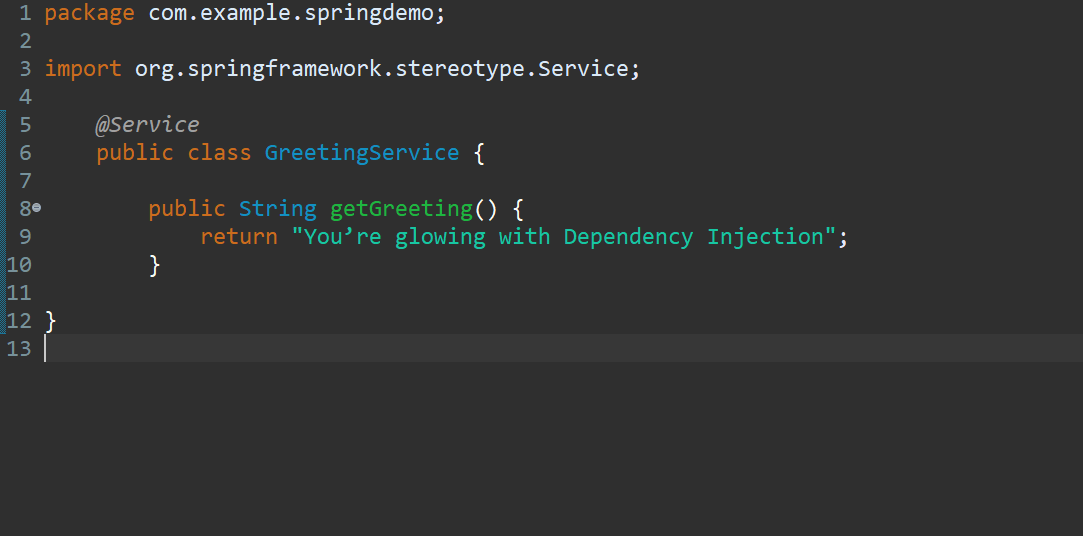
****

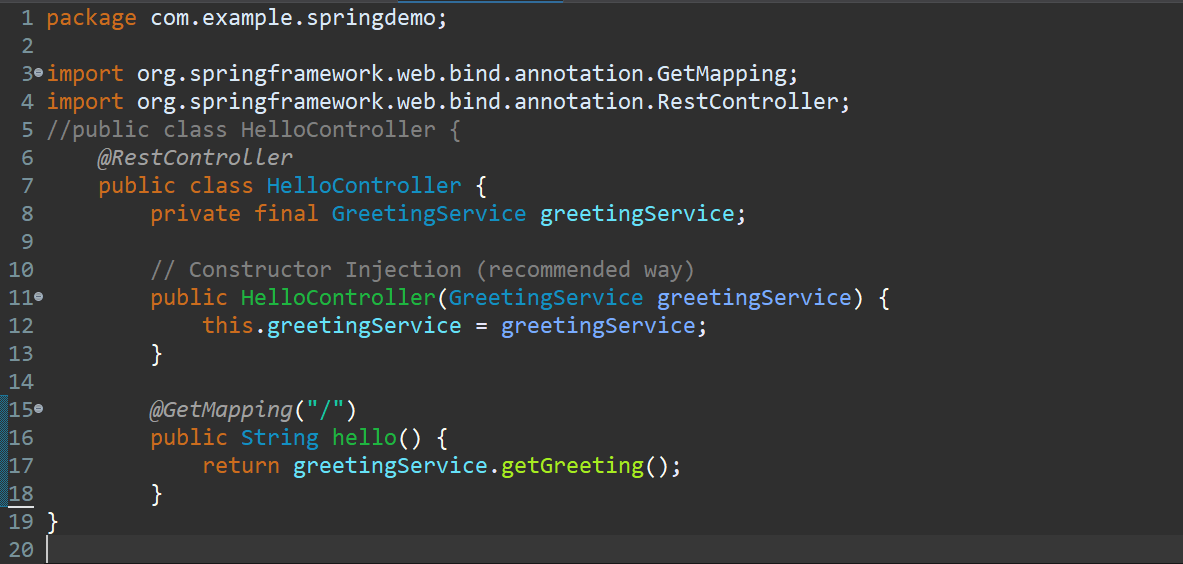
****

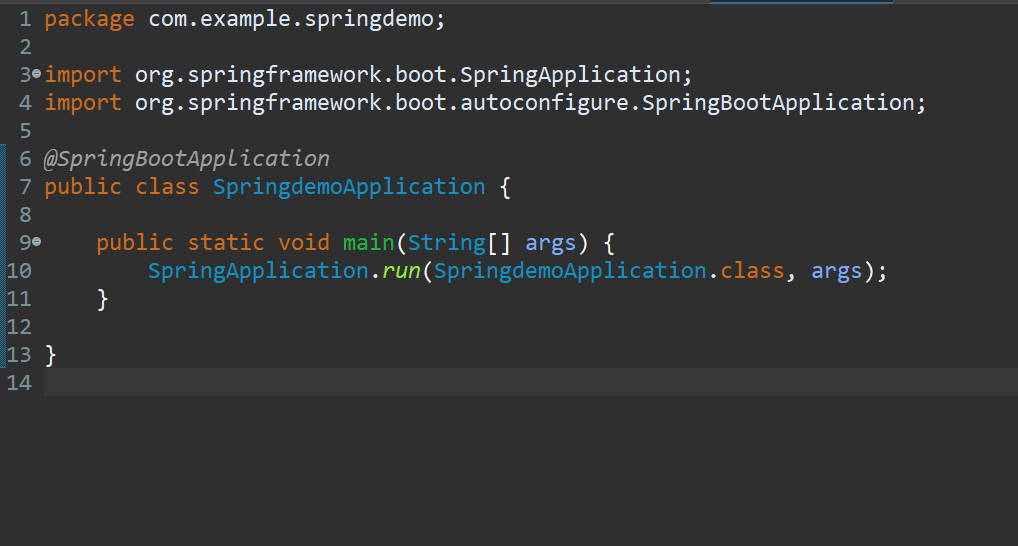
**Output:**

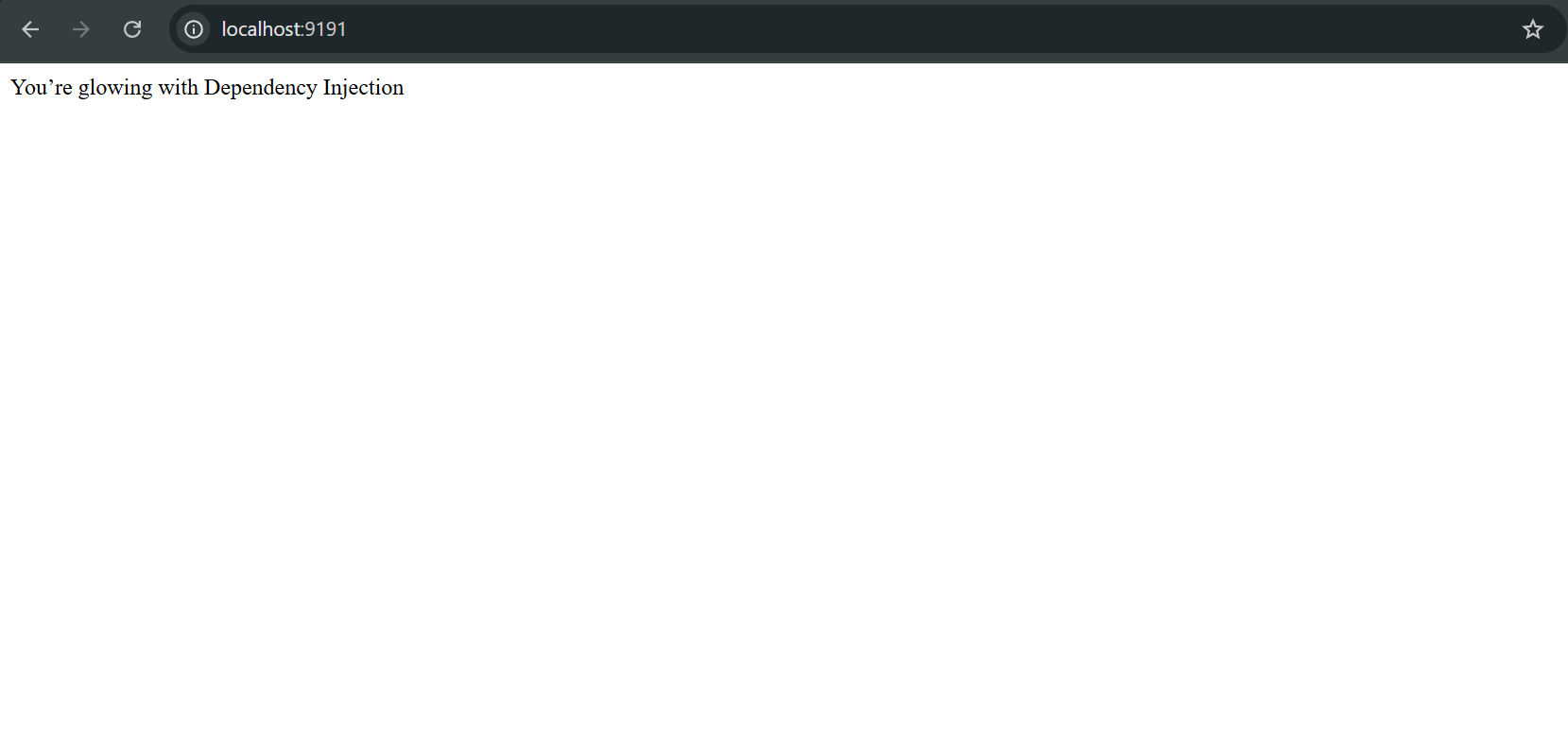
****

**Exercise 2: Implementing Dependency Injection**

****

****

****

****

**Exercise 4: Creating and Configuring a Maven Project**

**Step 1: Open eclipse**

1. **Go to:  
   File → New → Project...**
2. **Select:  
   Maven Project → Click Next**
3. **On the next screen:**
   * ***Uncheck* "Use default workspace location" if you want a custom location**
   * **Click Next**

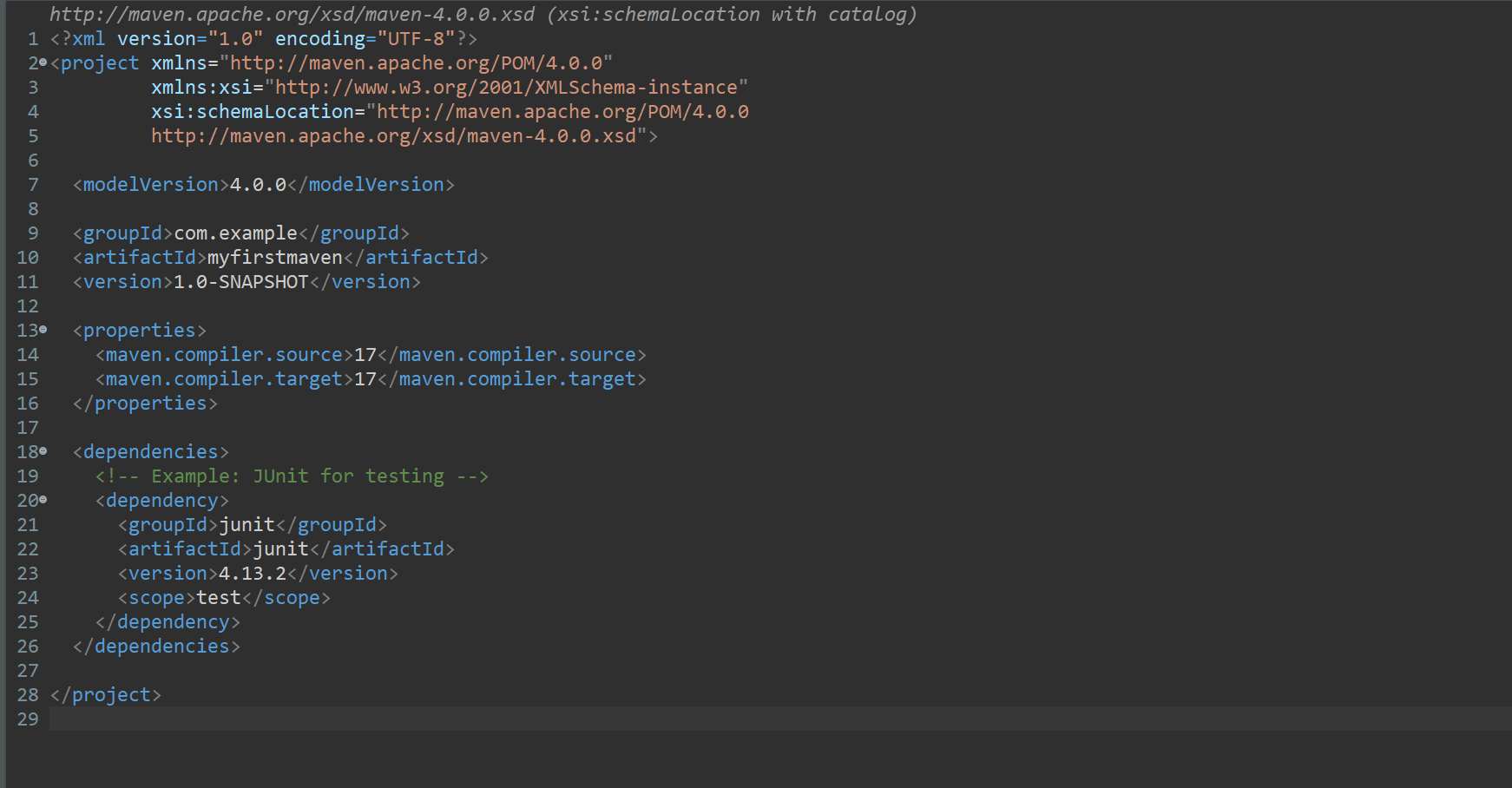
**Step 2: Archetype Selection**

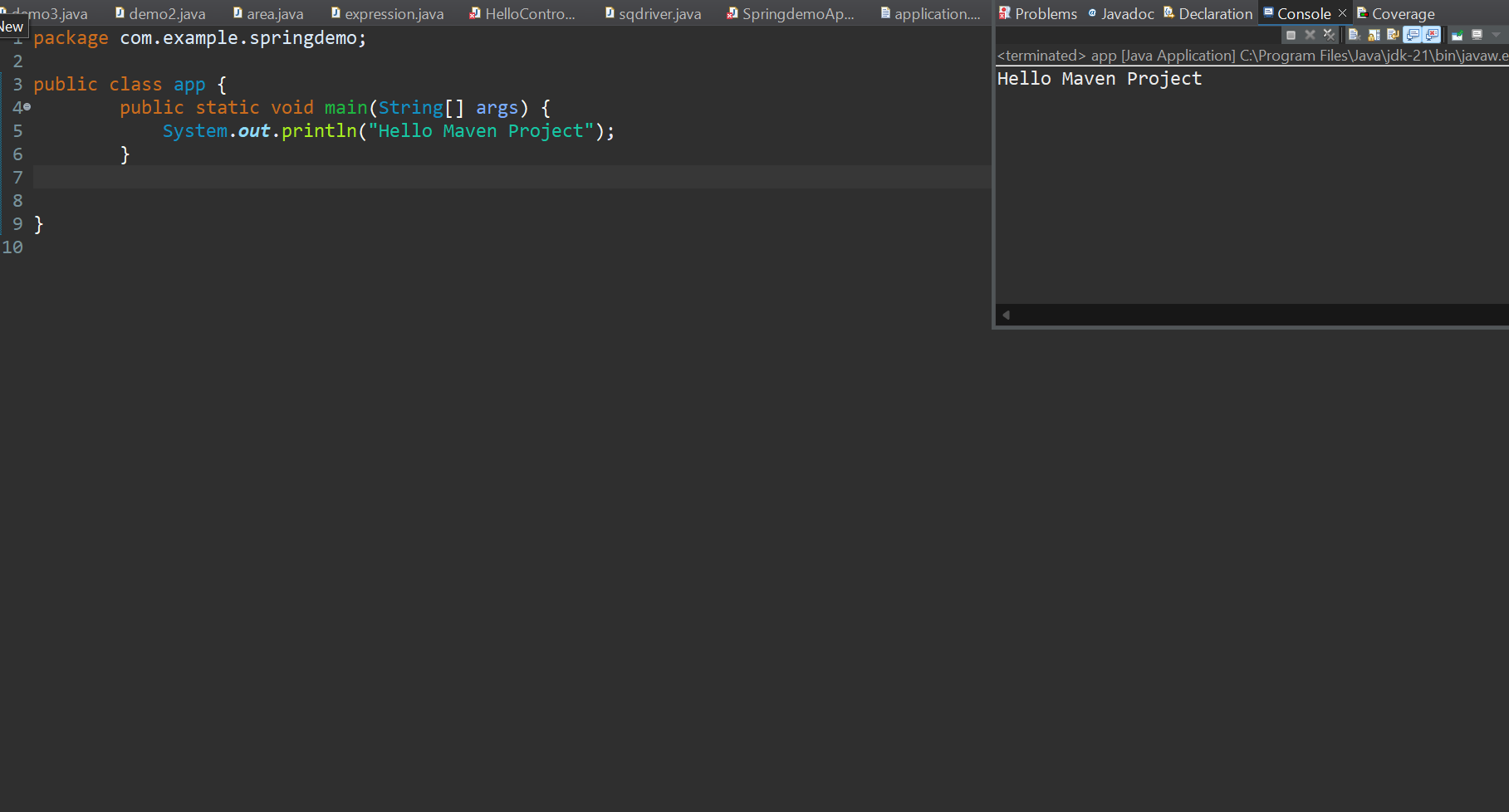
**You’ll now be asked to pick an archetype (a template for your project).**

1. **Choose: maven-archetype-quickstart**
2. **Click Next**

**Step 3: Enter Project Coordinates**

* **Group Id: com.example**
* **Artifact Id: myfirstmaven**
* **Version: Leave as default (1.0-SNAPSHOT)**
* **Click Finish**

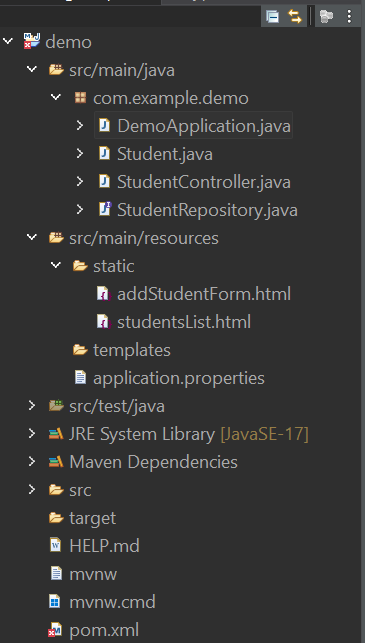
****

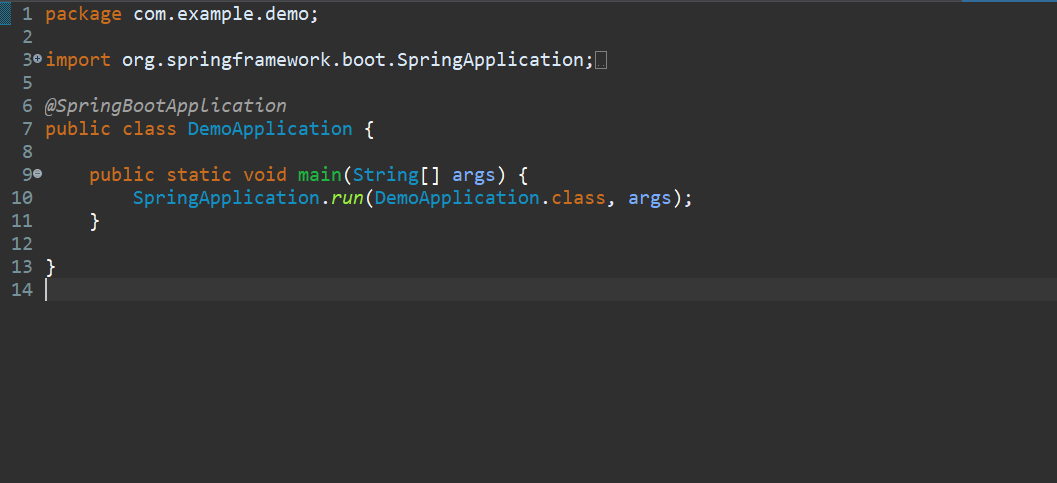
****

**SPRING- DATA-JPA-HANDSON**

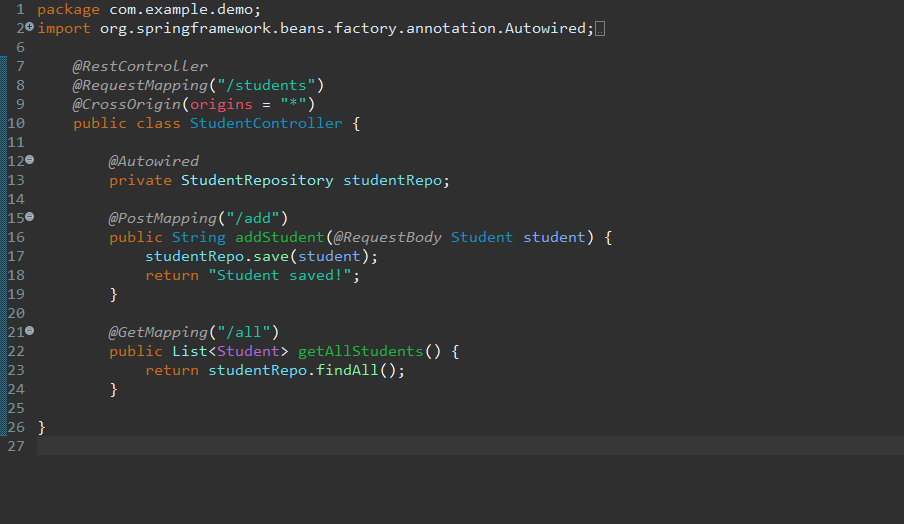
**SUPERSET ID :6410152**

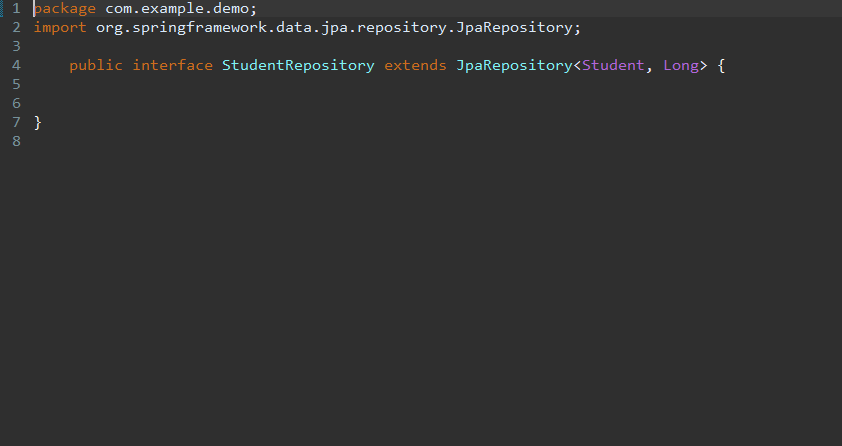
Spring Data JPA - Quick Example:

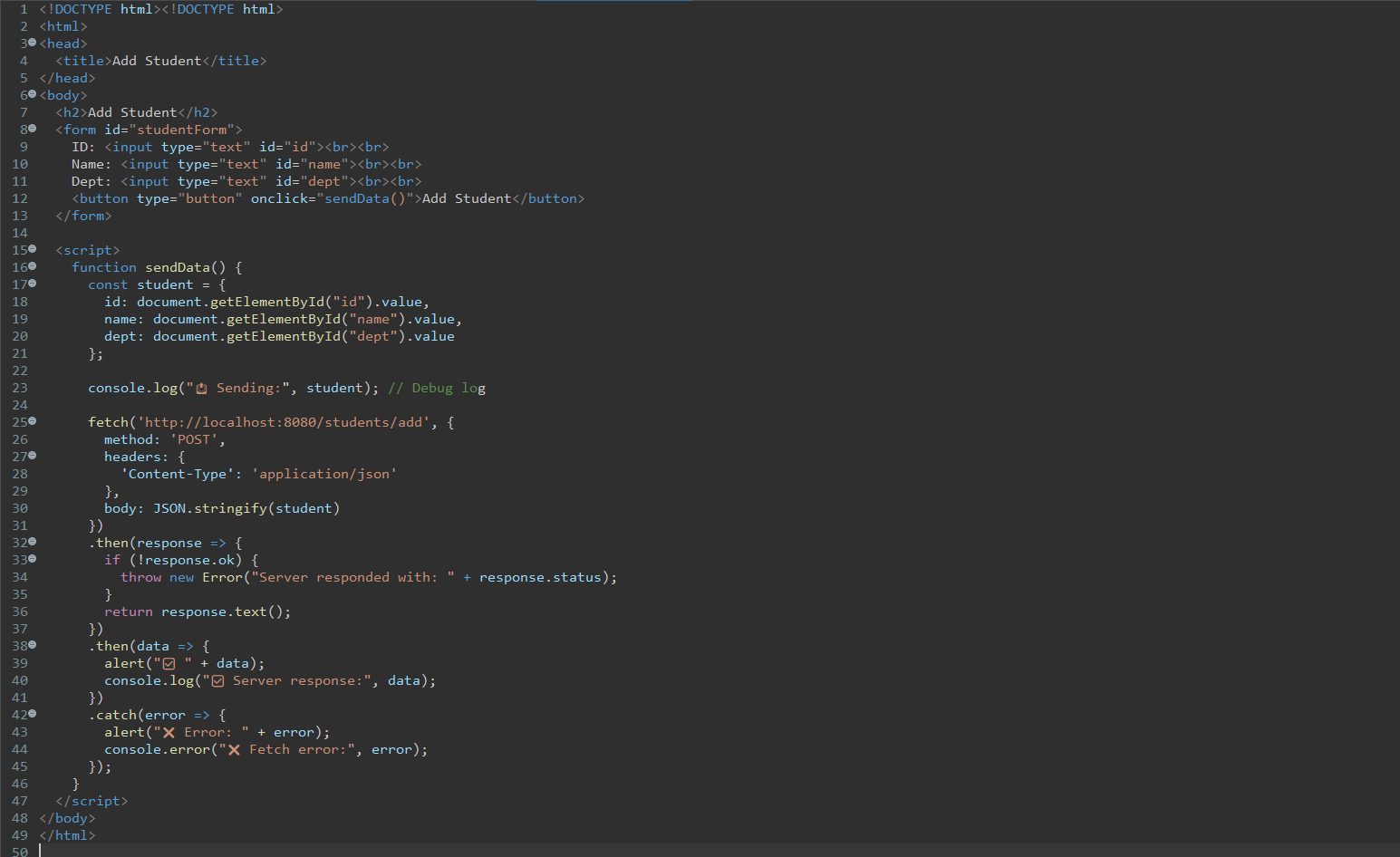


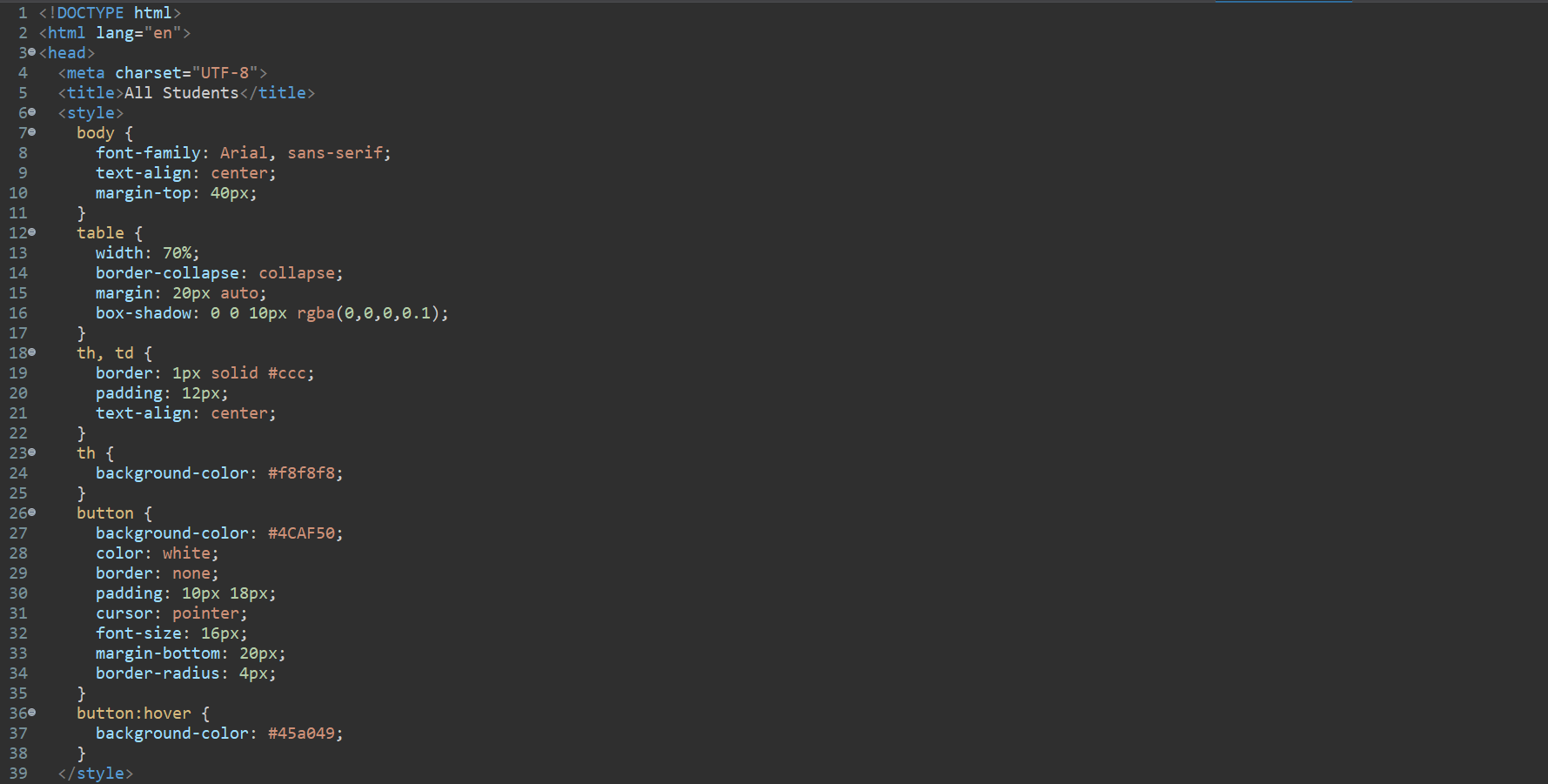


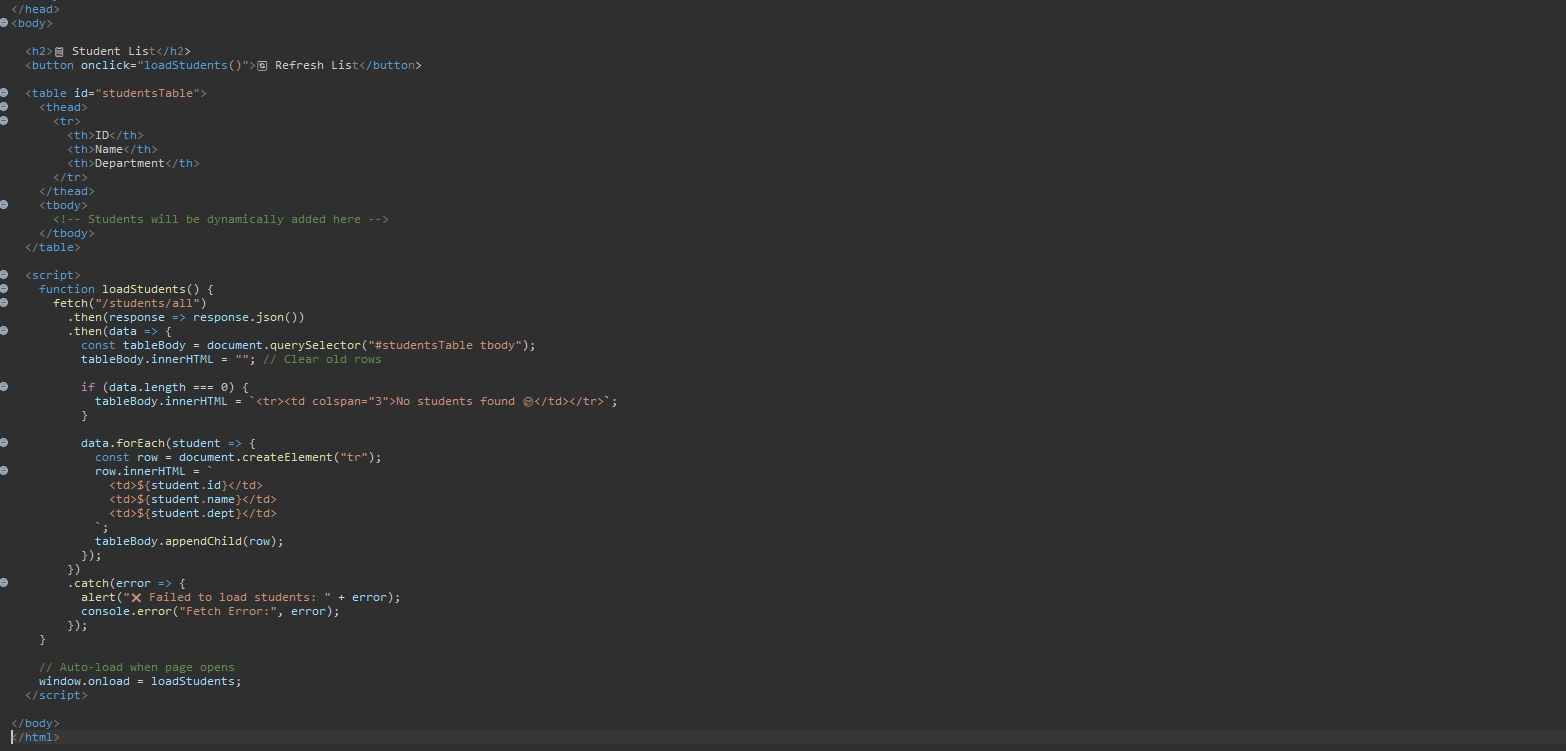
****

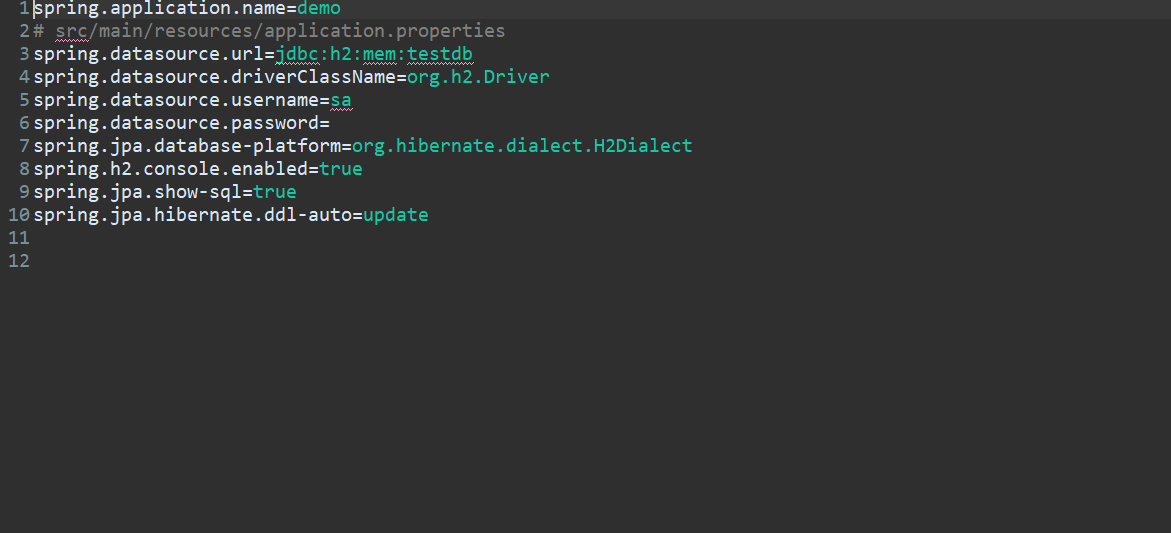
****

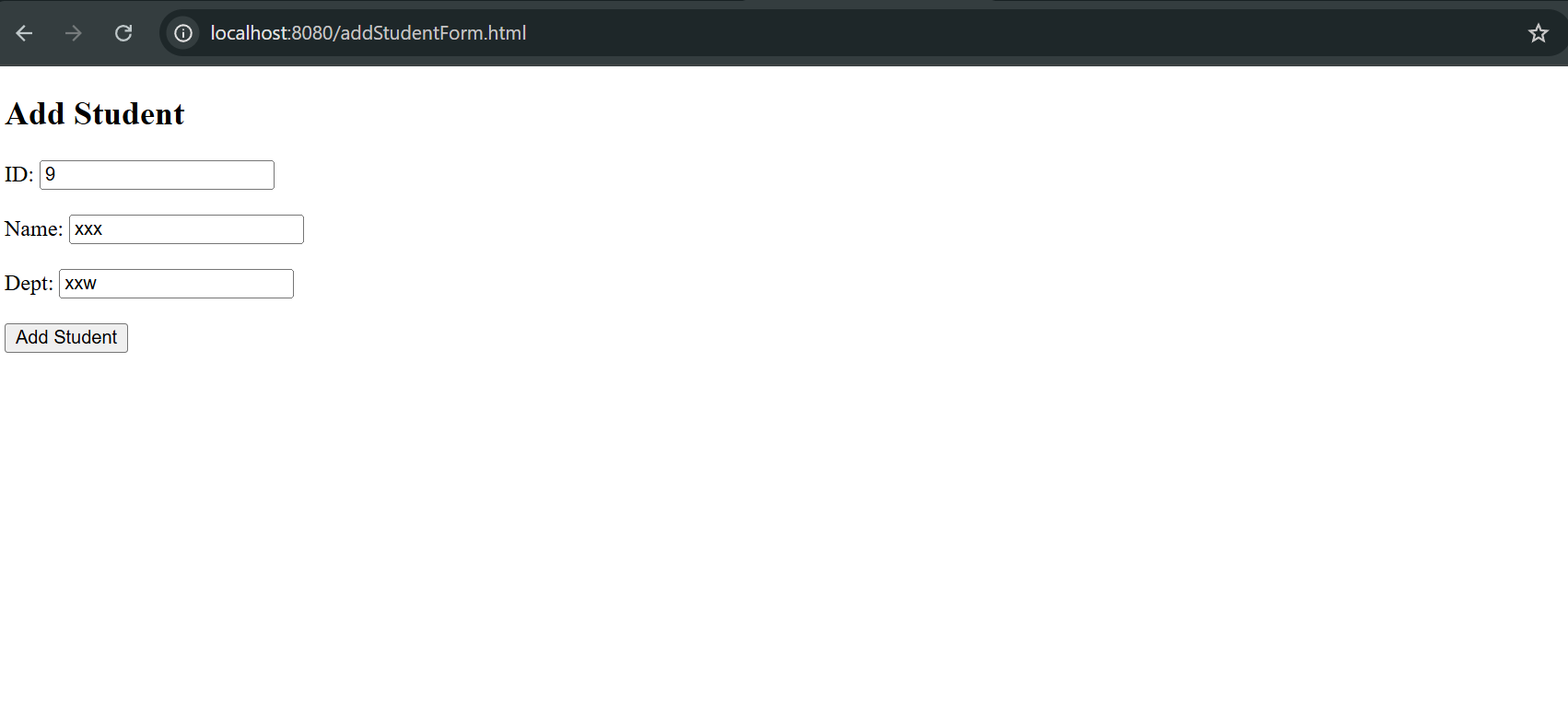
****

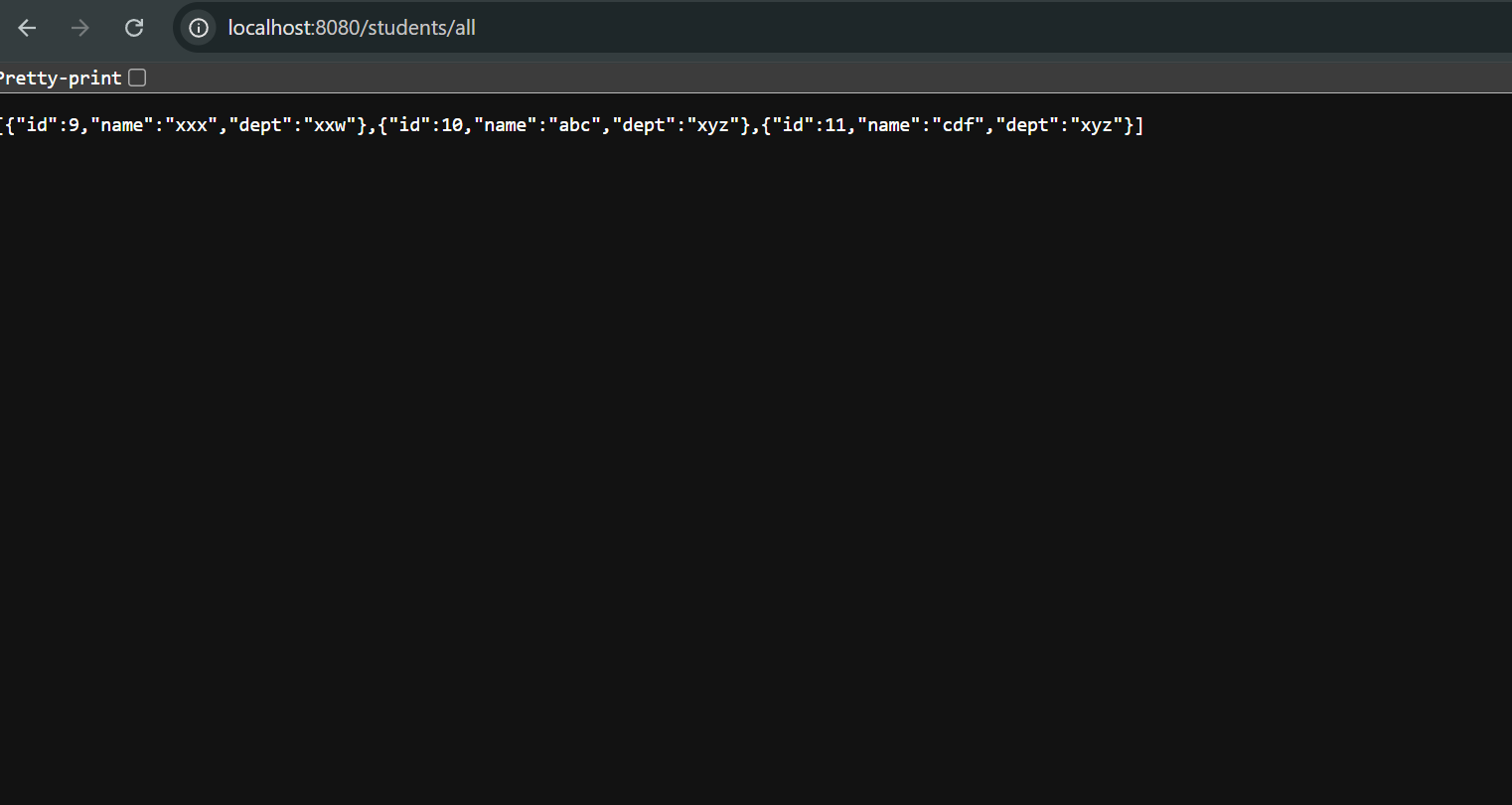
****

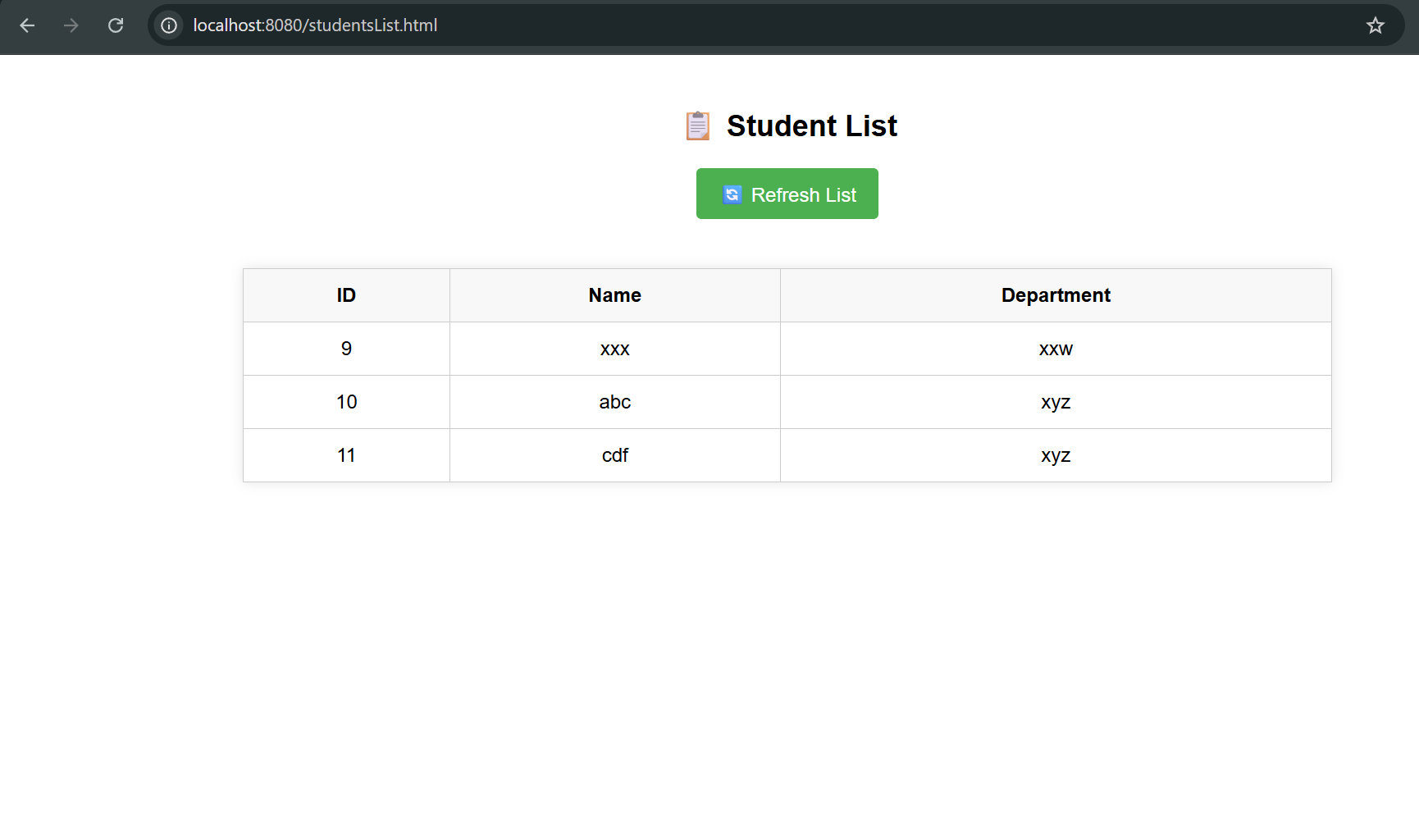
****

****

****

****

****

****

**Difference between JPA, Hibernate and Spring Data JPA:**

| **Feature/Aspect** | **JPA** | **Hibernate** | **Spring Data JPA** |
| --- | --- | --- | --- |
| **What it is** | **A *specification* (interface)** | **An *implementation* of JPA** | **A *framework on top* of JPA & Hibernate** |
| **Who provides it?** | **Oracle (part of Java EE/Jakarta EE)** | **Hibernate.org** | **Spring team (Pivotal)** |
| **Is it code or just idea?** | **Interface/abstract concept** | **Actual working code (implementation)** | **Working code with auto-implement magic** |
| **Boilerplate code?** | **A lot (you write repo logic)** | **Less, but still need DAO classes** | **Very little — Spring auto-generates repo logic** |
| **CRUD methods?** | **You write manually** | **You write in DAOs** | **Just extend JpaRepository, done** |
| **Learning curve** | **Medium** | **Medium** | **Super beginner-friendly** |
| **Example** | **@Entity, @Id** | **SessionFactory, @FetchType** | **StudentRepository extends JpaRepository** |
| **Usage** | **API standard** | **ORM tool** | **Abstraction over both** |