**Creating Microservices for Account and Loan**

In this hands-on exercise, we will create two independent Spring Boot microservices for a banking application—one for handling account information and another for handling loan details. Each microservice will be developed as a separate Maven-based Spring Boot application with its own pom.xml file and will run on a different port. These services are designed to be standalone with no backend or database integration.

**Step 1: Set Up Project Directory**

1. Open the D: drive on your system.
2. Create a new folder using your employee ID. For example:

E:\employee\_id

1. Inside that folder, create a directory named:

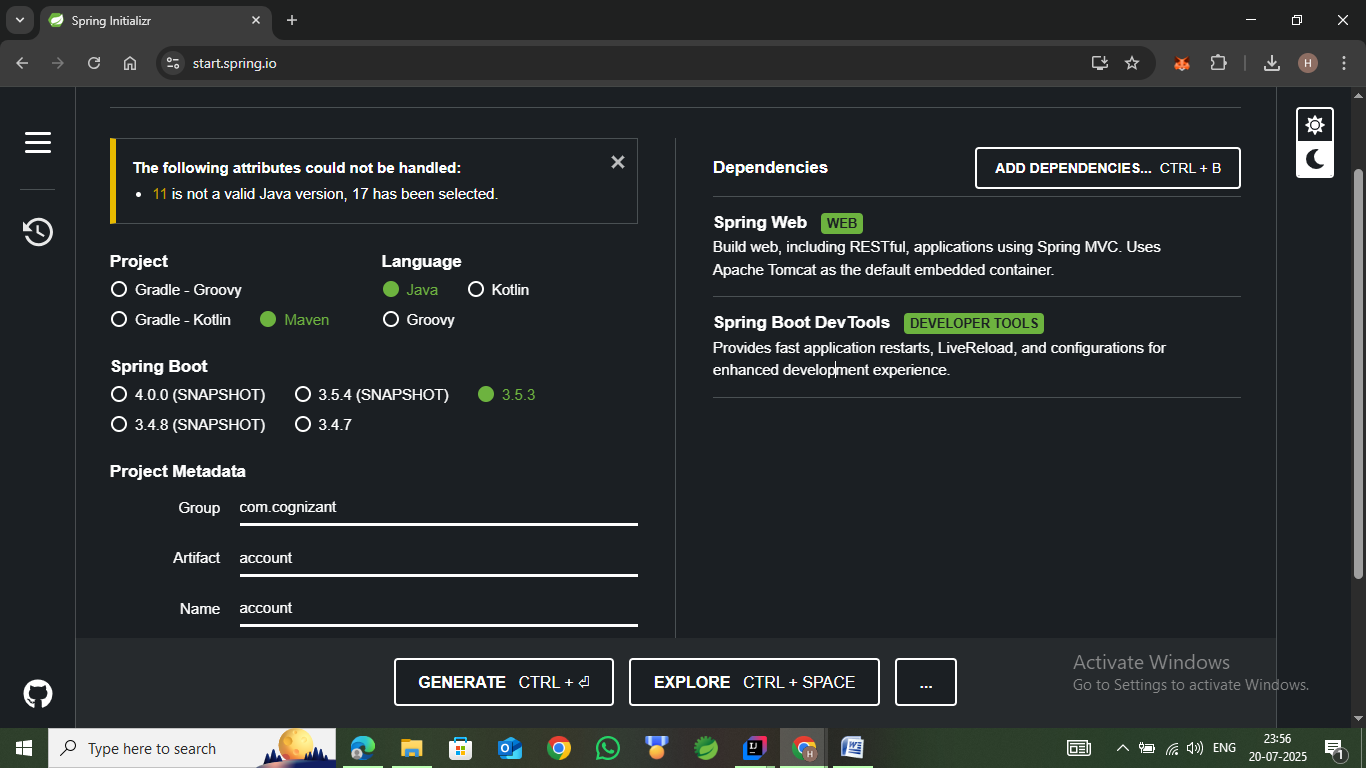
microservices

This will store both the account and loan microservice projects.

**Step 2: Create the Account Microservice**

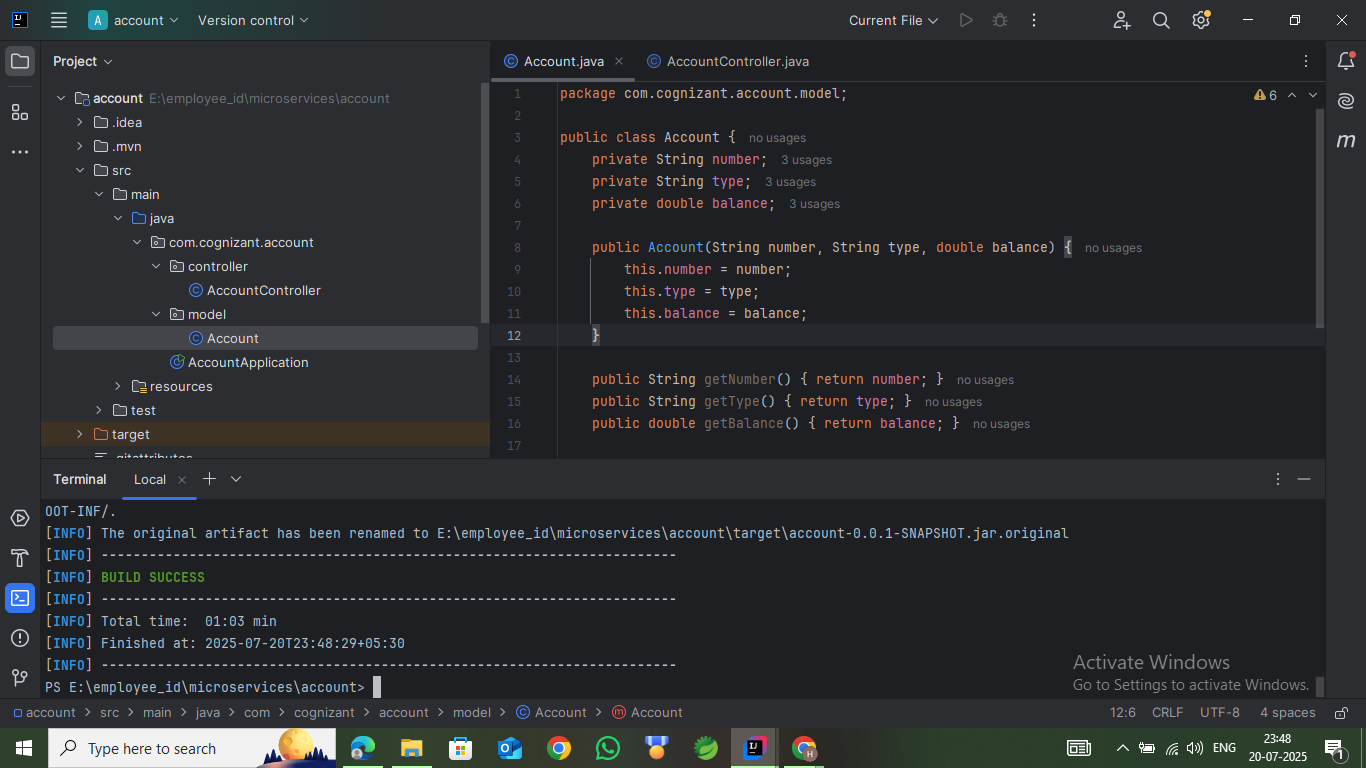
**2.1 Generate the Project**

1. Visit <https://start.spring.io>.
2. Fill in the following form fields:
   * Group: com.cognizant
   * Artifact: account
3. Add the following dependencies:
   * Spring Web
   * Spring Boot DevTools
4. Click on **Generate** to download the project as a ZIP file.

**2.2 Build the Project**

1. Extract the ZIP file.
2. Move the extracted account folder into the microservices directory.
3. Open Command Prompt in the account directory and run:

mvn clean package

****

**Step 3: Implement the Account Microservice**

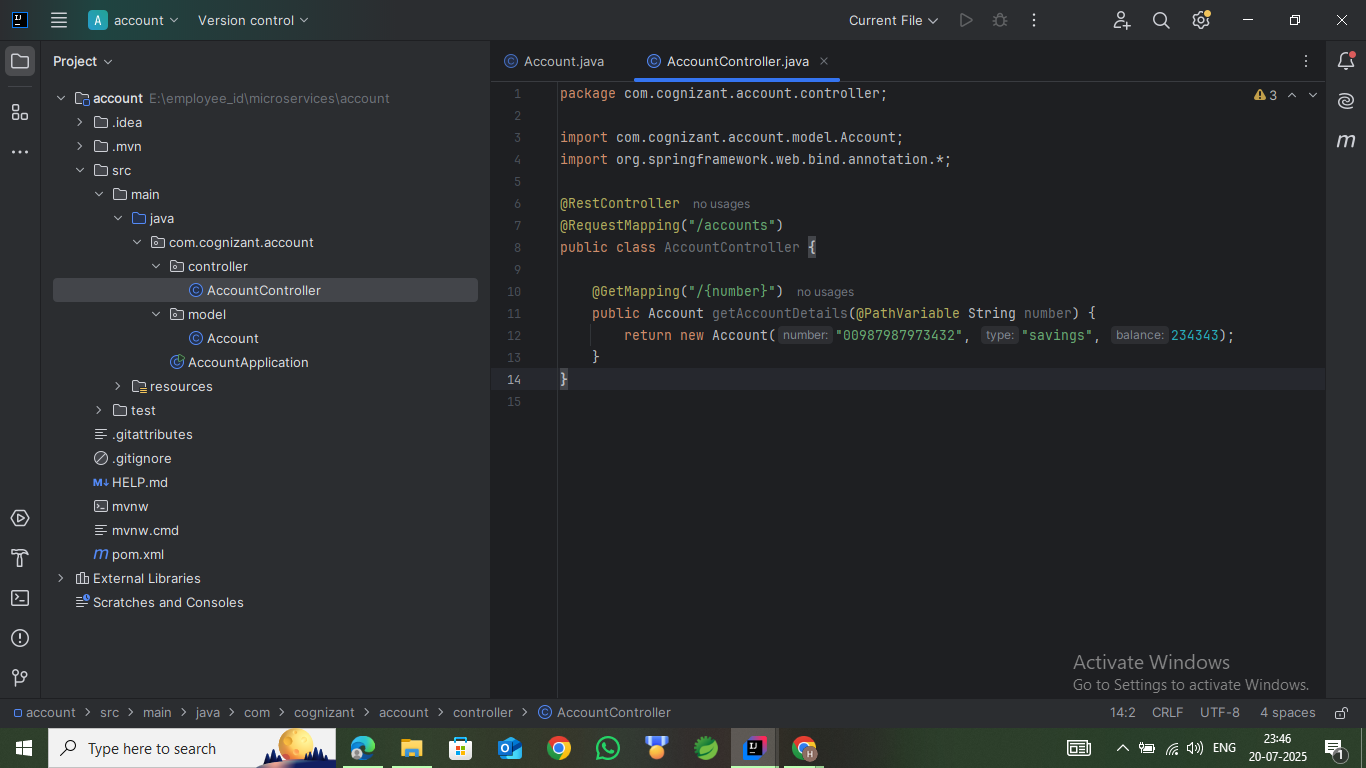
**3.1 Import Project into Eclipse**

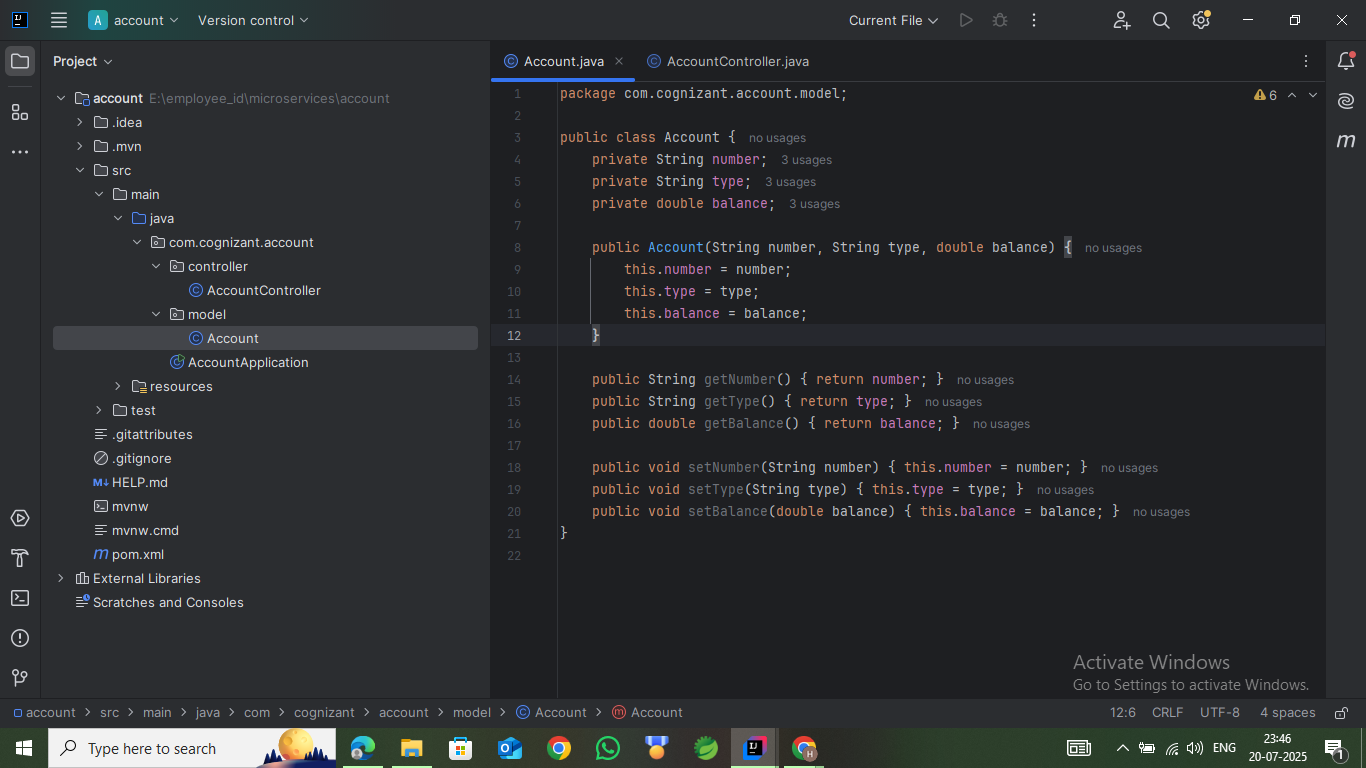
* Open Eclipse.
* Go to File → Import → Maven → Existing Maven Projects.
* Select the account project directory to import it.

**3.2 Create Model and Controller Classes**

Inside the src/main/java/com/cognizant/account directory, create the following packages and files:

* model/Account.java
* controller/AccountController.java

****

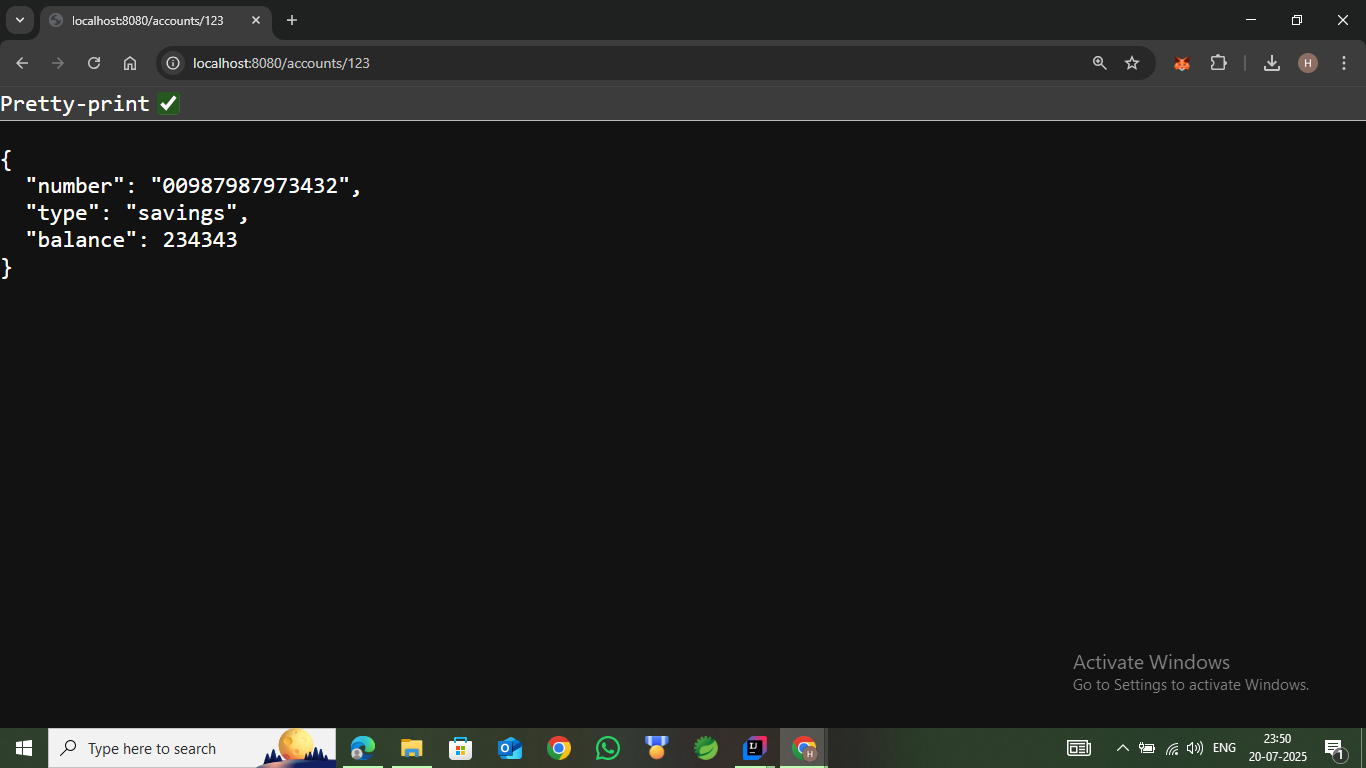
****

**Step 4: Run and Test the Account Microservice**

1. Run the AccountApplication class.
2. Open a browser or use Postman to test the endpoint:

http://localhost:8080/accounts/123

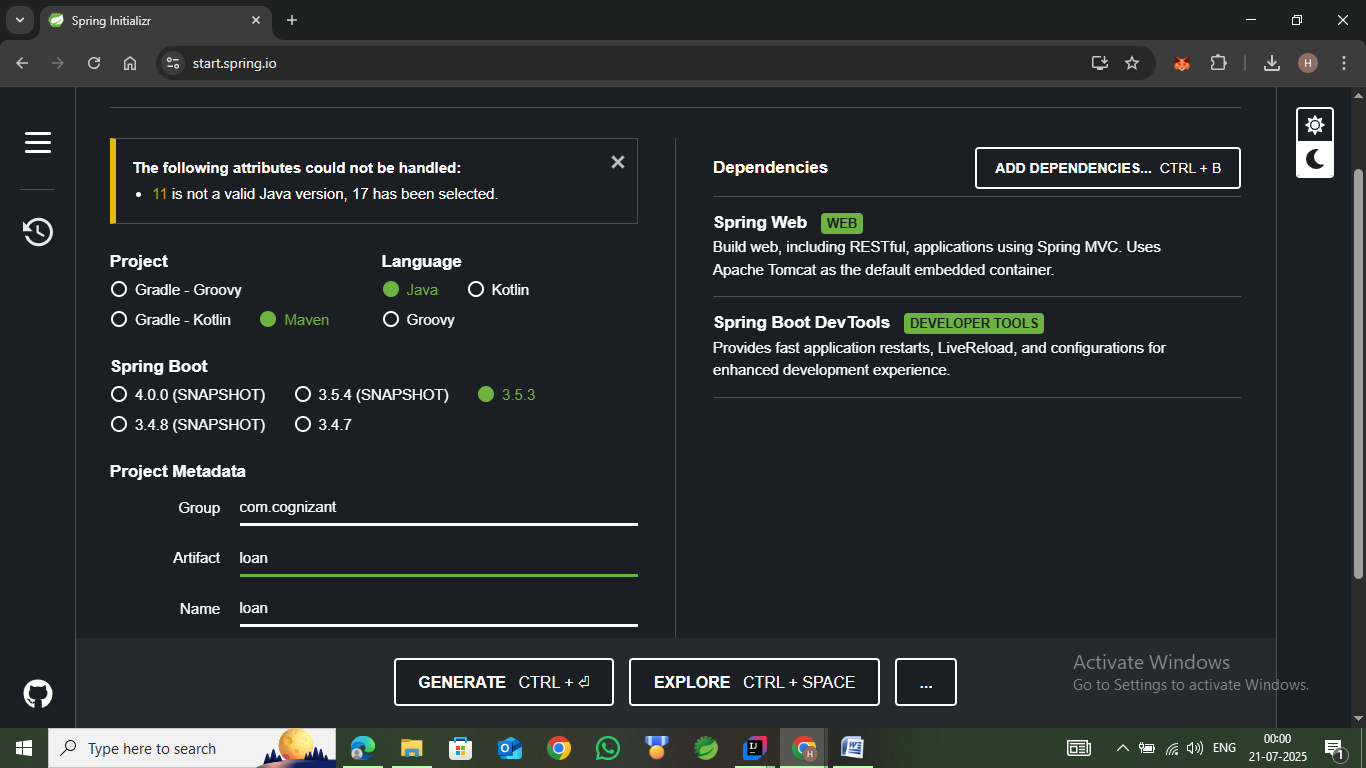
You should receive a dummy JSON response containing the account number, type, and balance.



**Step 5: Create the Loan Microservice**

**5.1 Generate the Project**

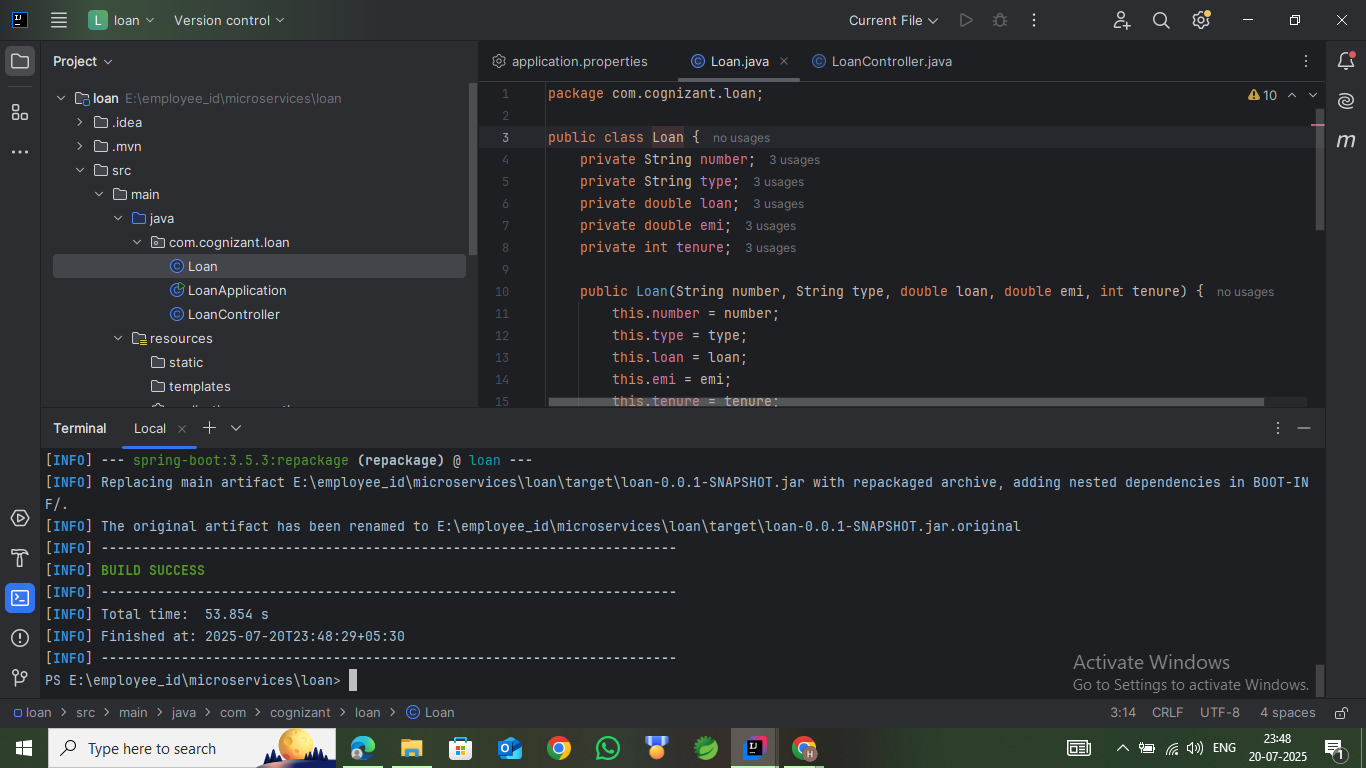
1. Visit <https://start.spring.io>.
2. Fill in the following form fields:
   * Group: com.cognizant
   * Artifact: loan
3. Add the following dependencies:
   * Spring Web
   * Spring Boot DevTools
4. Click on **Generate** to download the project.

****

**5.2 Build the Project**

1. Extract the ZIP file.
2. Move the extracted loan folder into the microservices directory.
3. Open Command Prompt in the loan folder and run:

mvn clean package

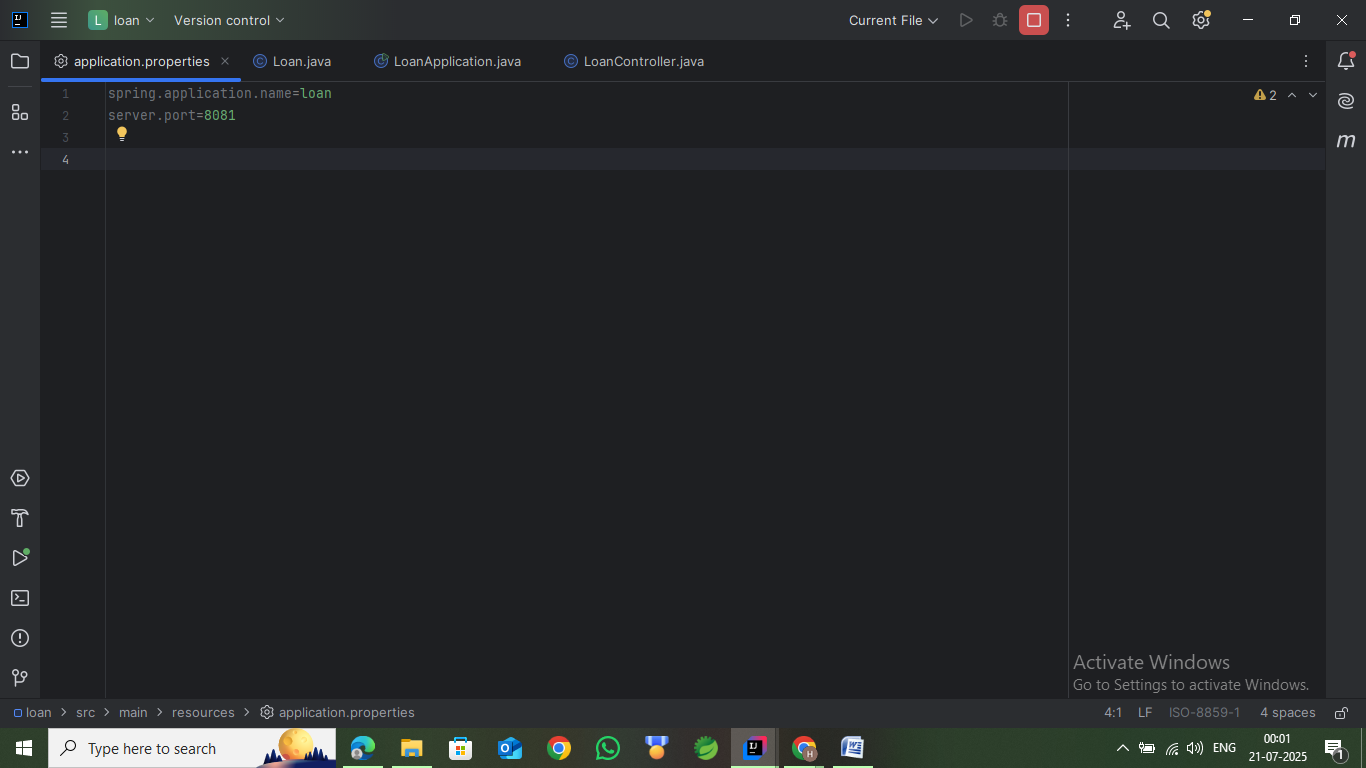


**Step 6: Configure Loan Service Port**

Since port 8080 is already in use by the Account service, the Loan service must run on a different port.

1. Open src/main/resources/application.properties.
2. Add the following line:

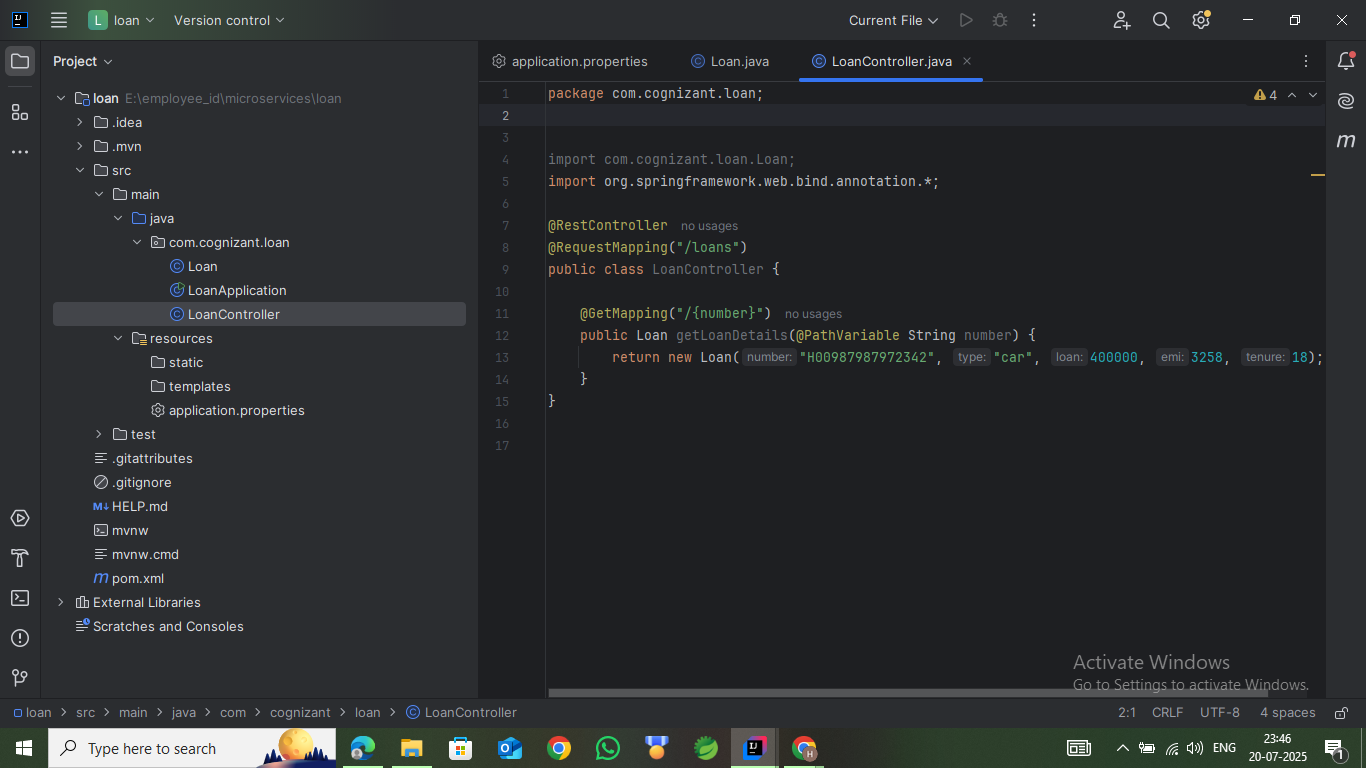
server.port=8081

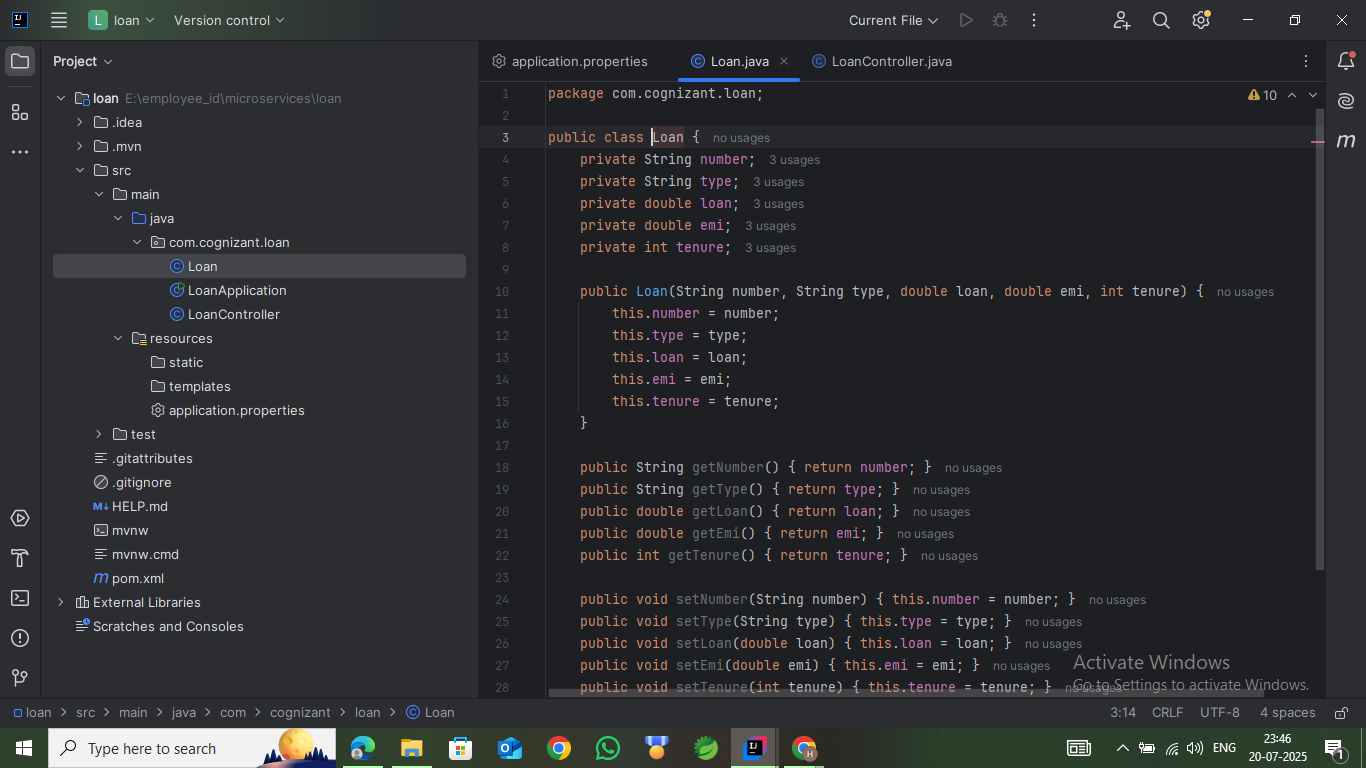
****

**Step 7: Implement the Loan Microservice**

Inside the src/main/java/com/cognizant/loan directory, create the following packages and files:

* Loan.java
* LoanController.java

****

****

**Step 8: Run and Test the Loan Microservice**

1. Make sure the Account microservice is still running on port 8080.
2. Run the LoanApplication class.
3. Open a browser or use Postman to test the loan endpoint:

http://localhost:8081/loans/456

You should receive a dummy JSON response with loan number, type, amount, EMI, and tenure.

