**Week 5 Hands-On**

**Creating Microservices for account and loan**

In this hands on exercises, we will create two microservices for a bank. One microservice for handing accounts and one for handling loans.

Each microservice will be a specific independent Spring RESTful Webservice maven project having it's own pom.xml. The only difference is that, instead of having both account and loan as a single application, it is split into two different applications. These webservices will be a simple service without any backend connectivity.

Follow steps below to implement the two microservices:

**Account Microservice**

1. Create a folder with your **employee ID** in the D: drive.
2. Create a folder named microservices inside the folder created in the previous step.  
   This folder will contain all the sample projects that we will create for learning microservices.
3. Open <https://start.spring.io/> in a browser.
4. Enter the following values in the form:
   * **Group**: com.cognizant
   * **Artifact**: account
5. Select the following modules:
   * Developer Tools → Spring Boot DevTools
   * Web → Spring Web
6. Click **Generate** and download the ZIP file.
7. Extract the account folder from the ZIP and place it inside the microservices folder created earlier.
8. Open Command Prompt in the account folder and build the project using:

go

mvn clean package

1. Import this project into Eclipse.
2. Implement a controller method for **getting account details** based on account number:
   * **Method**: GET
   * **Endpoint**: /accounts/{number}
   * **Sample Response** (just a dummy response without any backend connectivity):

json

{

"number": "00987987973432",

"type": "savings",

"balance": 234343

}

1. Launch the application by running the main application class and test the service in the browser.

**Loan Microservice**

1. Follow similar steps as mentioned above for the **Account Microservice**.
2. Implement a service API to **get loan account details**:
   * **Method**: GET
   * **Endpoint**: /loans/{number}
   * **Sample Response** (just a dummy response without any backend connectivity):

json

{

"number": "H00987987972342",

"type": "car",

"loan": 400000,

"emi": 3258,

"tenure": 18

}

1. Launch this application **after having the account service already running**.
2. This launch will **fail initially** with an error:  
   "Bind address is already in use"
3. Reason: Each service runs on default port 8080, which is already in use by the Account service.
4. To fix this, include the following in application.properties:

ini

server.port=8081

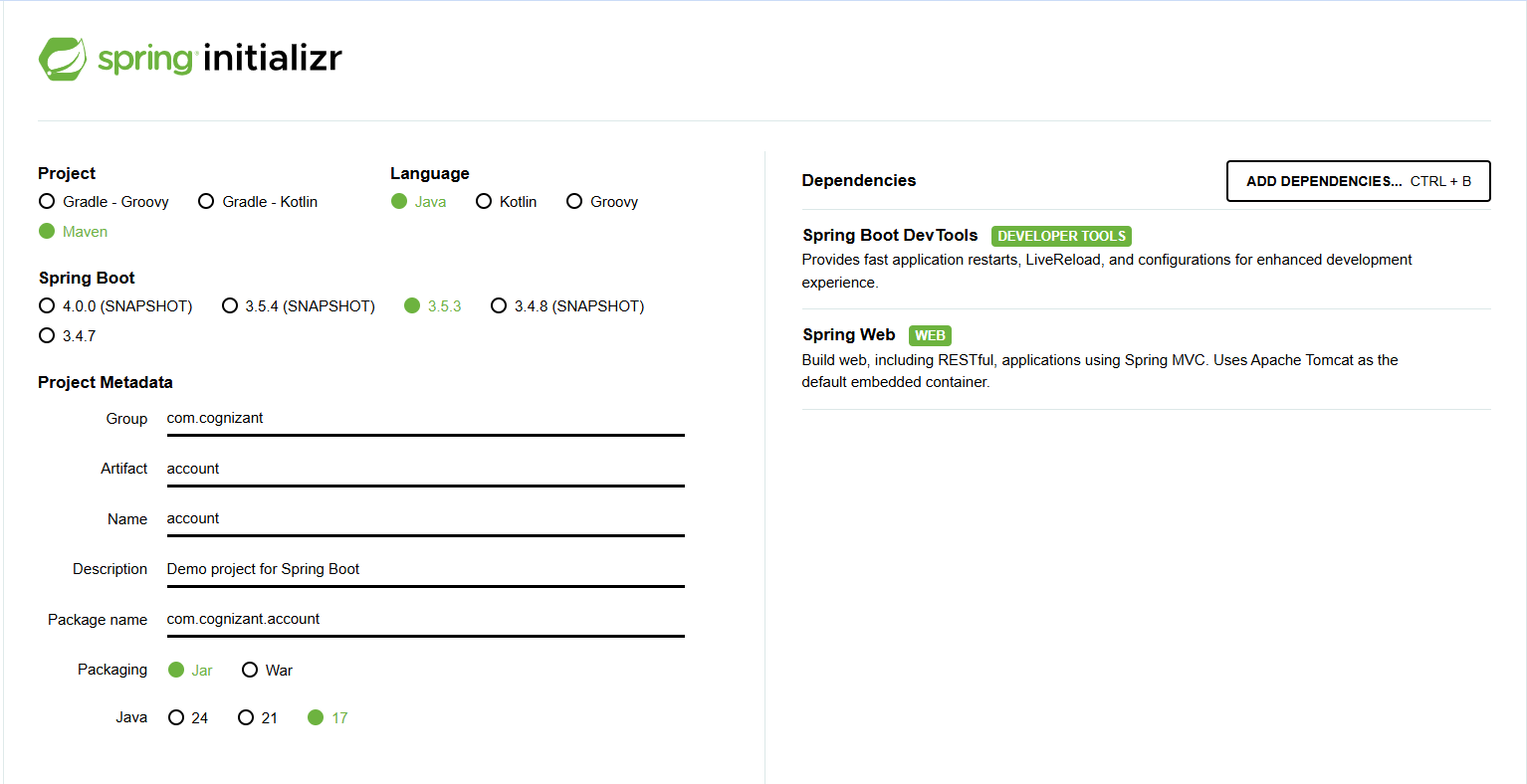
1. Relaunch the Loan service application.
2. Test the Loan service on:

http://localhost:8081/loans/{number}

**Eclipse Console Note**

* The **console window in Eclipse** will have both service logs running.
* To switch between different consoles, use the **monitor icon** in the Eclipse console view.

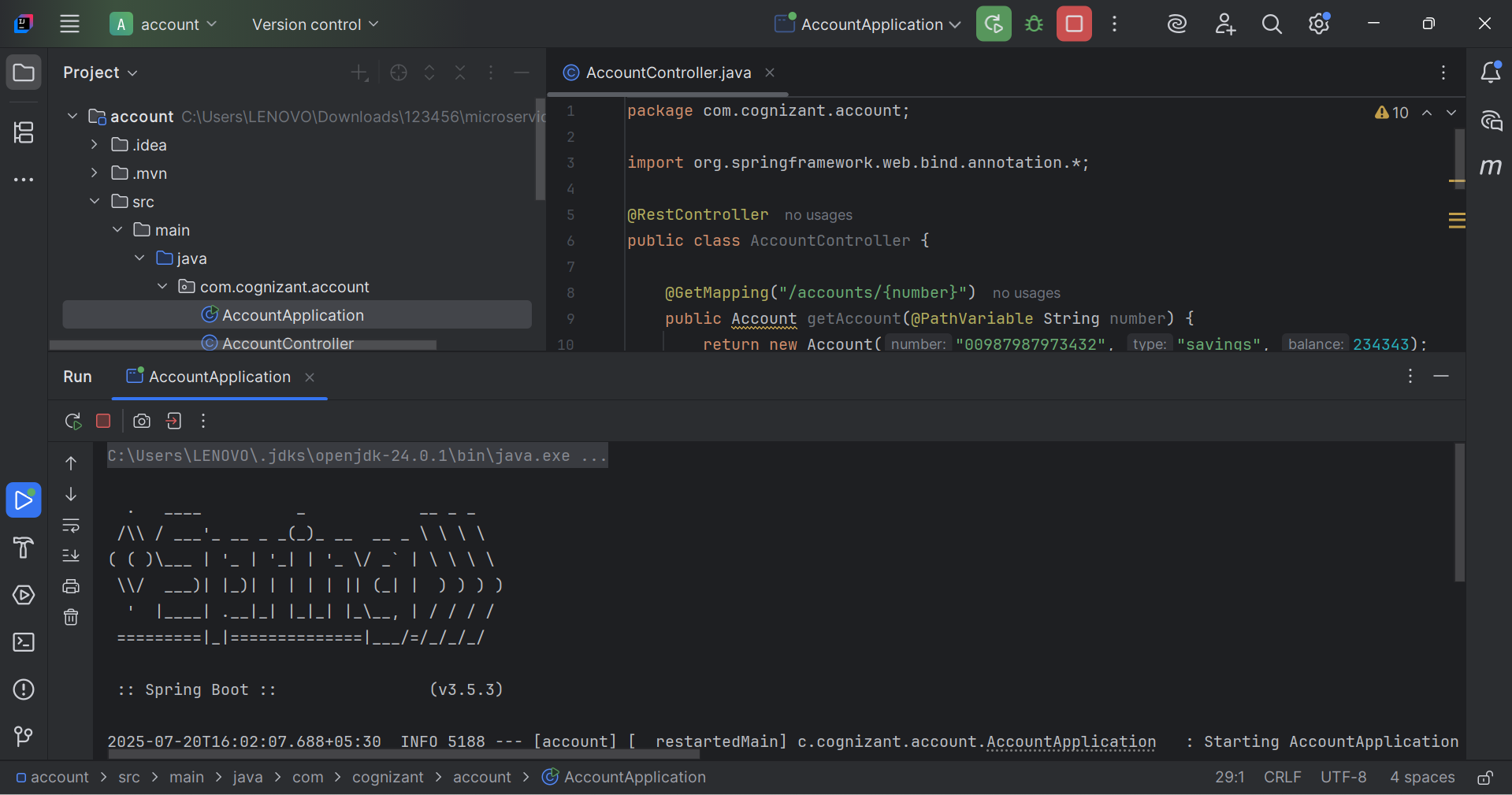
**Account Microservice**

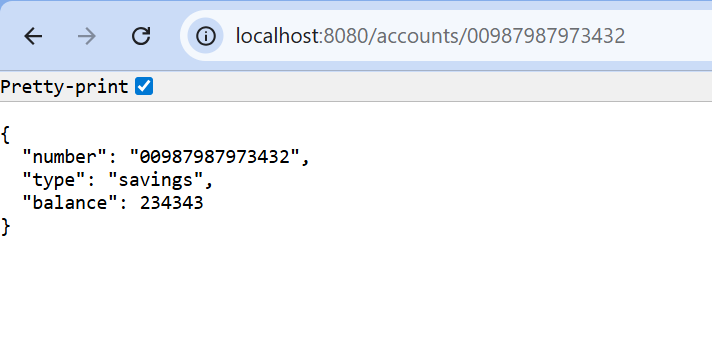
****

AccountController.java

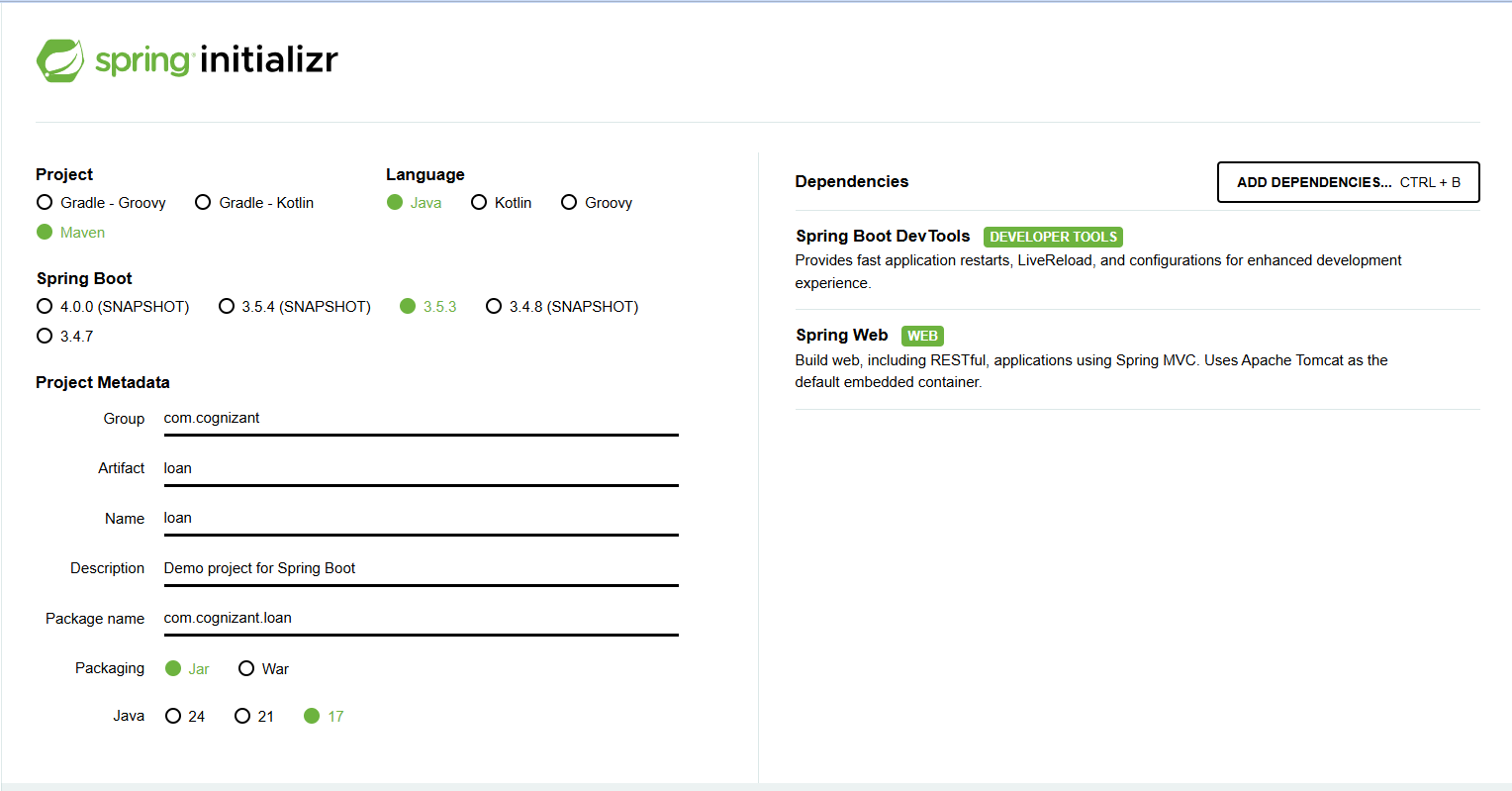
package com.cognizant.account;  
  
import org.springframework.web.bind.annotation.\*;  
  
@RestController  
public class AccountController {  
  
 @GetMapping("/accounts/{number}")  
 public Account getAccount(@PathVariable String number) {  
 return new Account("00987987973432", "savings", 234343);  
 }  
  
 static class Account {  
 private String number;  
 private String type;  
 private double balance;  
  
 public Account(String number, String type, double balance) {  
 this.number = number;  
 this.type = type;  
 this.balance = balance;  
 }  
  
 public String getNumber() { return number; }  
 public String getType() { return type; }  
 public double getBalance() { return balance; }  
 }  
}

**Output:**

****

****

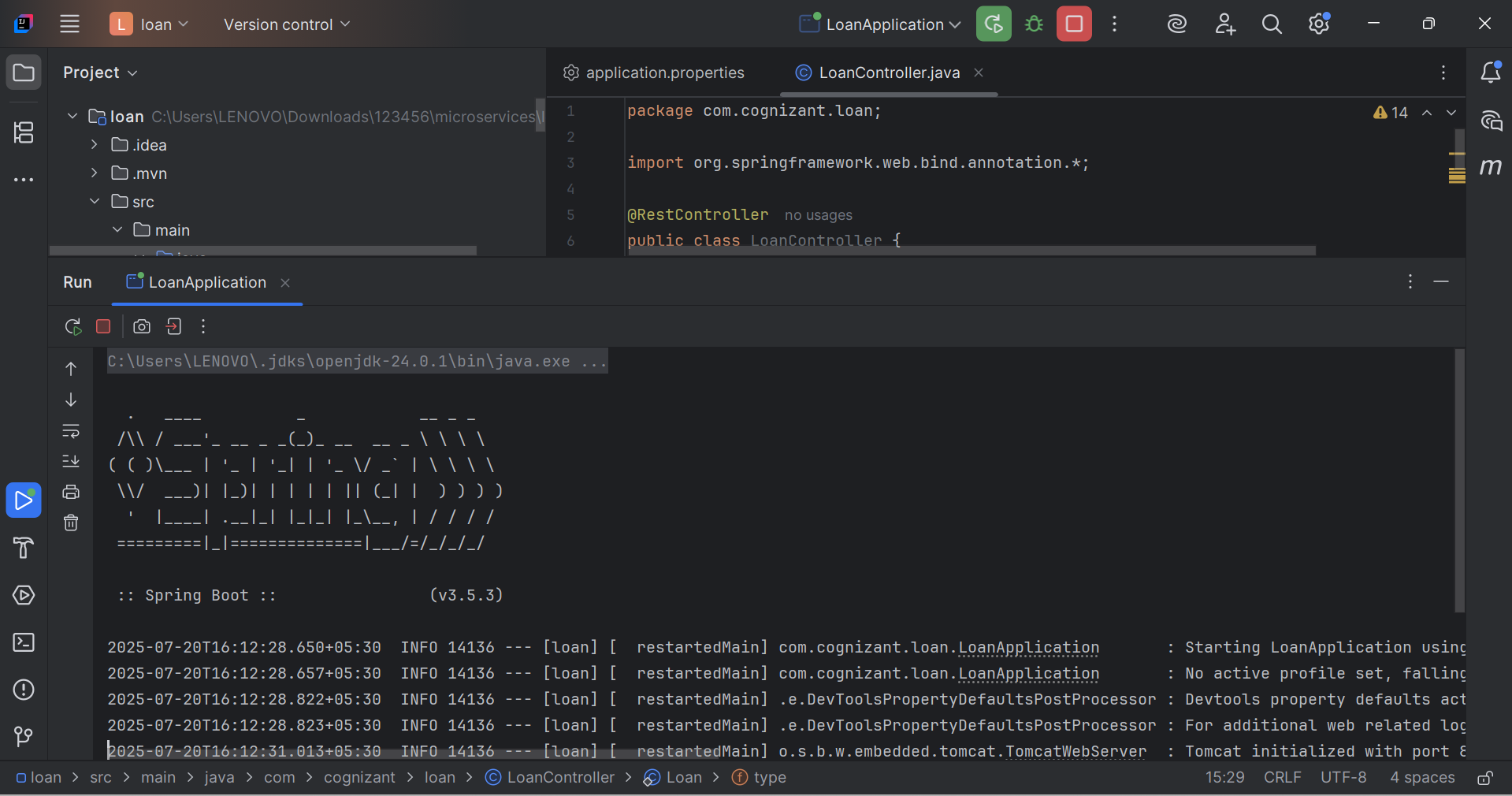
**Loan Microservice**

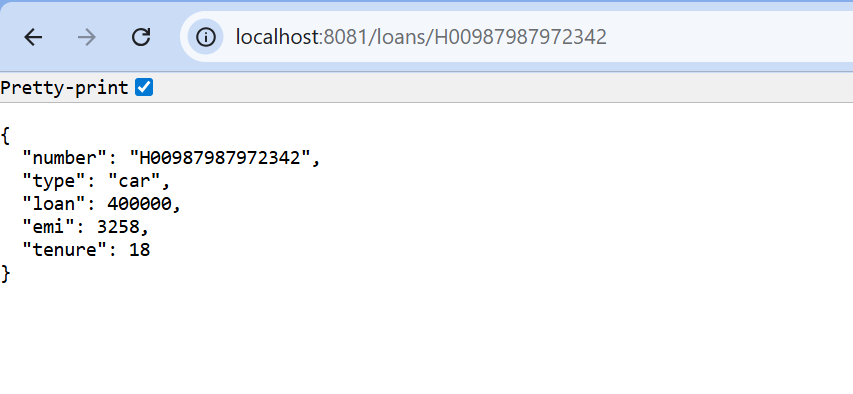
****

LoanController.java

package com.cognizant.loan;  
  
import org.springframework.web.bind.annotation.\*;  
  
@RestController  
public class LoanController {  
  
 @GetMapping("/loans/{number}")  
 public Loan getLoan(@PathVariable String number) {  
 return new Loan("H00987987972342", "car", 400000, 3258, 18);  
 }  
  
 static class Loan {  
 private String number;  
 private String type;  
 private double loan;  
 private double emi;  
 private int tenure;  
  
 public Loan(String number, String type, double loan, double emi, int tenure) {  
 this.number = number;  
 this.type = type;  
 this.loan = loan;  
 this.emi = emi;  
 this.tenure = tenure;  
 }  
  
 public String getNumber() { return number; }  
 public String getType() { return type; }  
 public double getLoan() { return loan; }  
 public double getEmi() { return emi; }  
 public int getTenure() { return tenure; }  
 }  
}

**Output:**

****

****