**MICROSERVICES HANDSON**

**Ex:1 Creating Microservices for account and loan  
  
1. Account Microservice:**

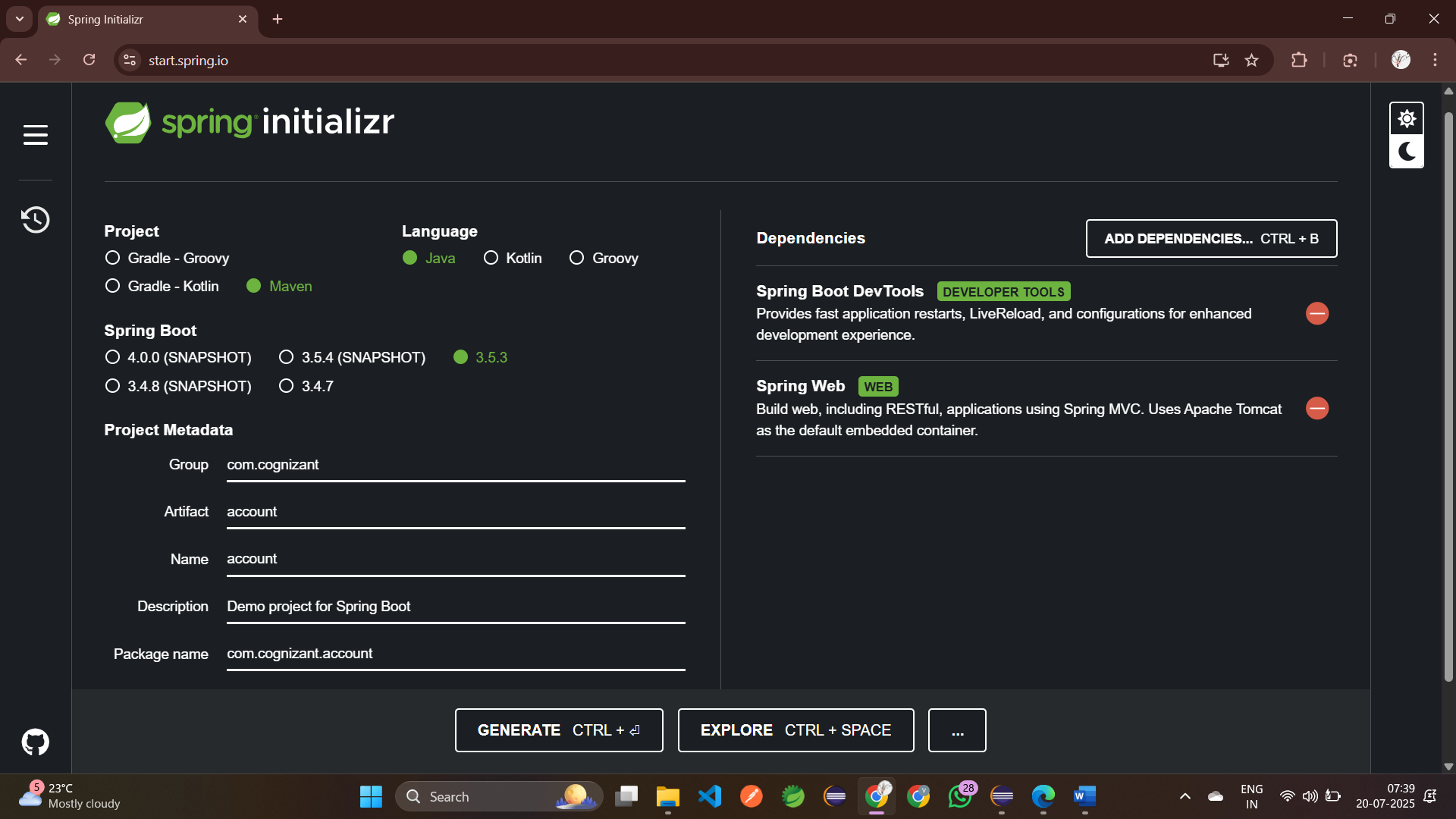
• Create folder with employee id in D: drive

• Create folder named 'microservices' in the new folder created in previous step. This folder will contain all the sample projects that we will create for learning microservices.

• Open https://start.spring.io/ in browser

• Enter form field values as specified below:   
 Group: com.cognizant   
 Artifact: account

• Select the following modules   
 Developer Tools > Spring Boot DevTools   
 Web > Spring Web

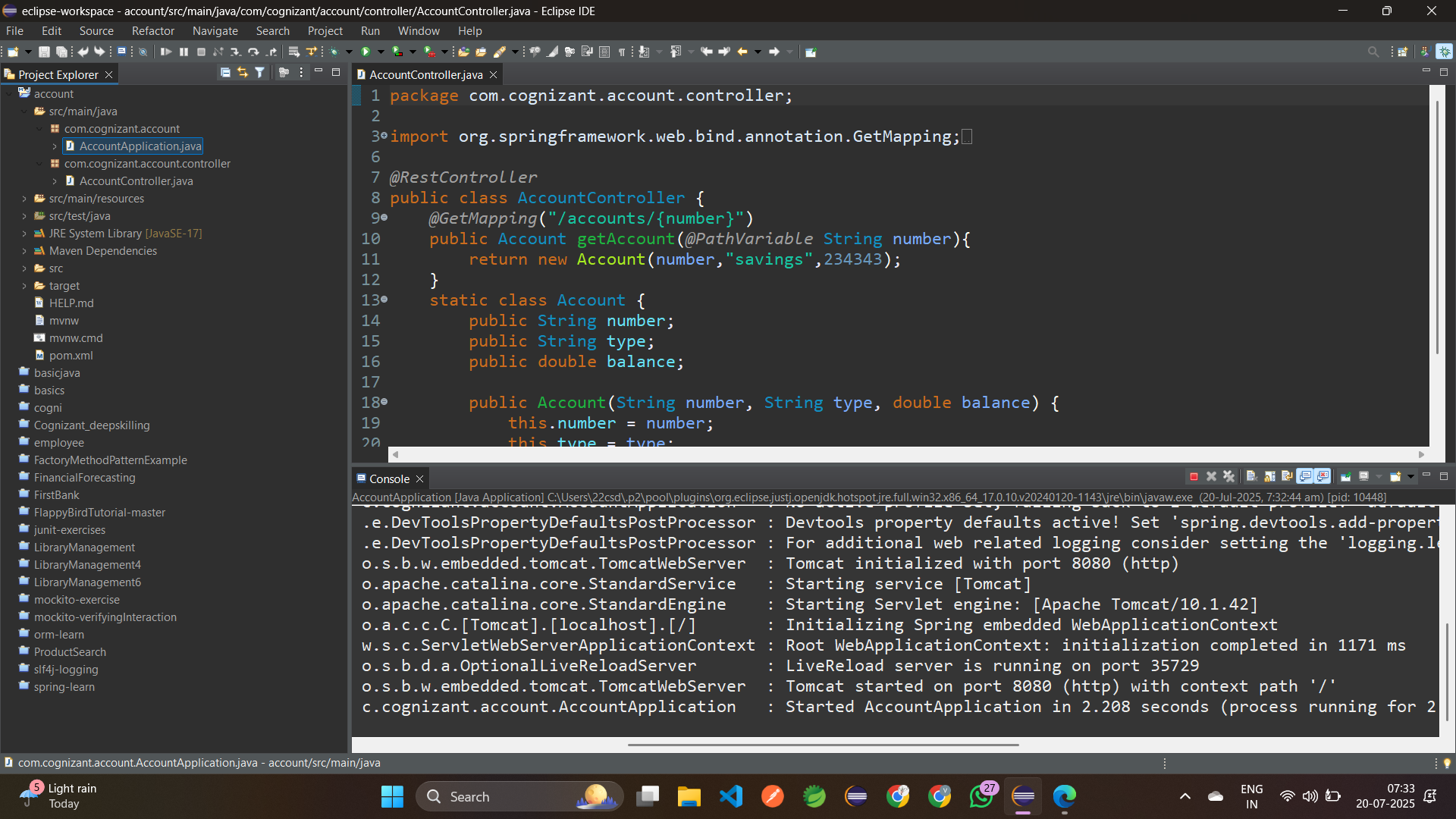
• Click generate and download the zip file   
  


• Extract 'account' folder from the zip and place this folder in the 'microservices' folder created earlier

• Open command prompt in account folder and build using mvn clean package command

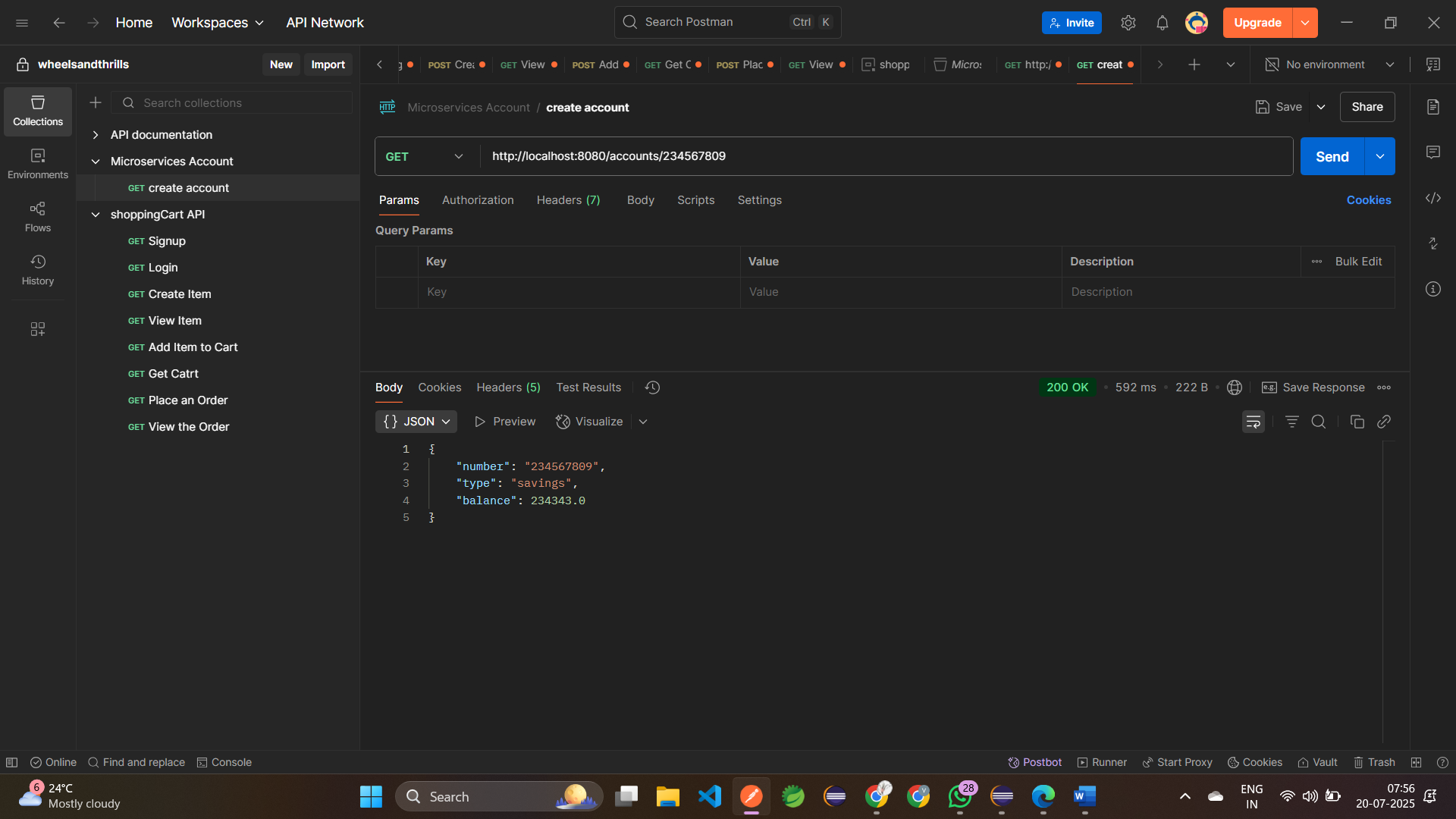
• Import this project in Eclipse and implement a controller method for getting account details based on account number.   
  
  
  
 1. Creating a Controller Package in **com.cognizant.account.controller**  
 2. Next, create a Controller class in java named **AccountContoller.java  
 Method:** GET Endpoint: /accounts/{number}  
 package com.cognizant.account.controller;  
 import org.springframework.web.bind.annotation.GetMapping;  
 import org.springframework.web.bind.annotation.PathVariable;  
 import org.springframework.web.bind.annotation.RestController;  
 @RestController  
 public class AccountController {  
 @GetMapping("/accounts/{number}")  
 public Account getAccount(@PathVariable String number){  
 return new Account(number,"savings",234343);  
 }  
 static class Account {  
 public String number;  
 public String type;  
 public double balance;  
 public Account(String number, String type, double balance) {  
 this.number = number;  
 this.type = type;  
 this.balance = balance;  
 }  
 }  
 }

3. Run the Application: **AccountApplication.java**



4. Testing:

i) Open postman  
 ii) Type the url: <http://localhost:8080/accounts/234567809>



**2. Loan Microservice**

• Follow similar steps specified for Account Microservice and implement a service API to get loan account details

• Open https://start.spring.io/ in browser

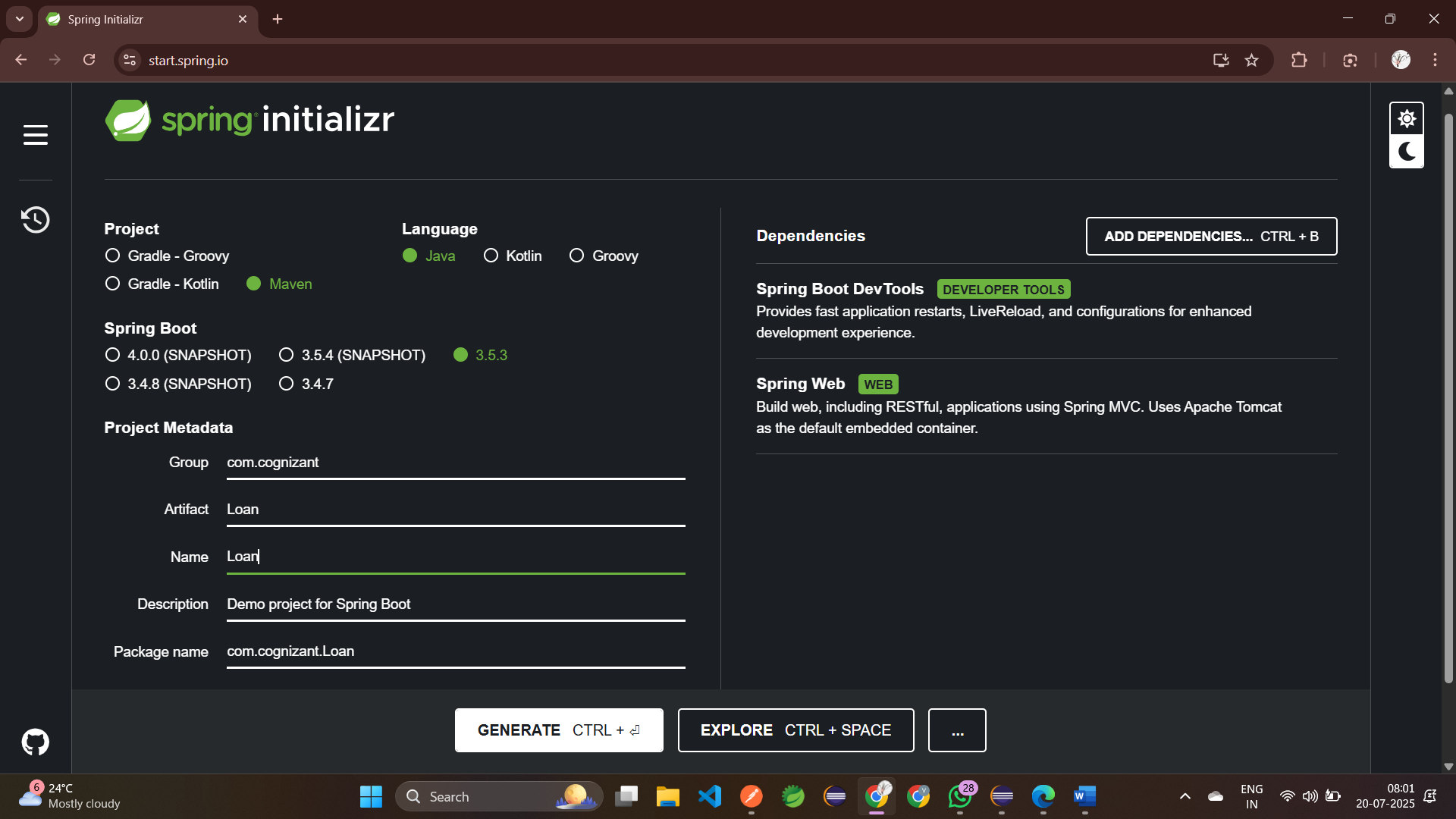
• Enter form field values as specified below:   
 Group: com.cognizant   
 Artifact: Loan

• Select the following modules   
 Developer Tools > Spring Boot DevTools   
 Web > Spring Web

• Click generate and download the zip file

• Extract 'account' folder from the zip and place this folder in the 'microservices' folder created earlier

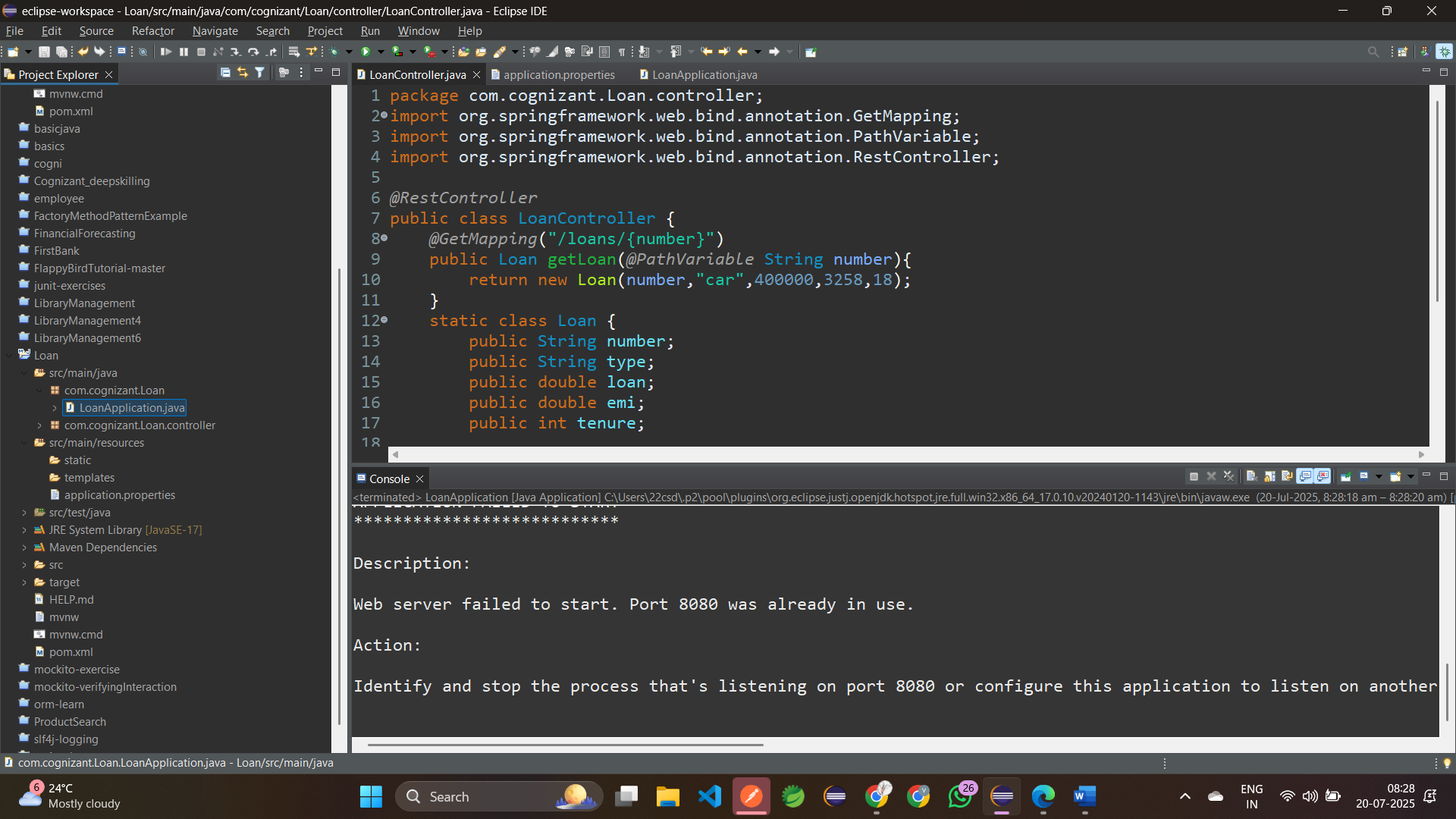
• Open command prompt in account folder and build using mvn clean package command



• Import this project in Eclipse and implement a controller method for getting account details based on account number.

1. Creating a Controller Package in **com.cognizant.Loan.controller**  
 2. Next, create a Controller class in java named **LoanContoller.java  
 Method:** GET Endpoint: /Loan/{number}  
 package com.cognizant.Loan.controller;  
 import org.springframework.web.bind.annotation.GetMapping;  
 import org.springframework.web.bind.annotation.PathVariable;  
 import org.springframework.web.bind.annotation.RestController;  
 @RestController  
 public class LoanController {  
 @GetMapping("/accounts/{number}")  
 public Loan getLoan(@PathVariable String number){  
 return new Loan(number,"savings",234343);  
 } static class Loan {  
 public String number;  
 public String type;  
 public double loan;  
 public double emi;  
 public int tenure;  
 public Account(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; }}}

3. Run the Application: **LoanApplication.java**

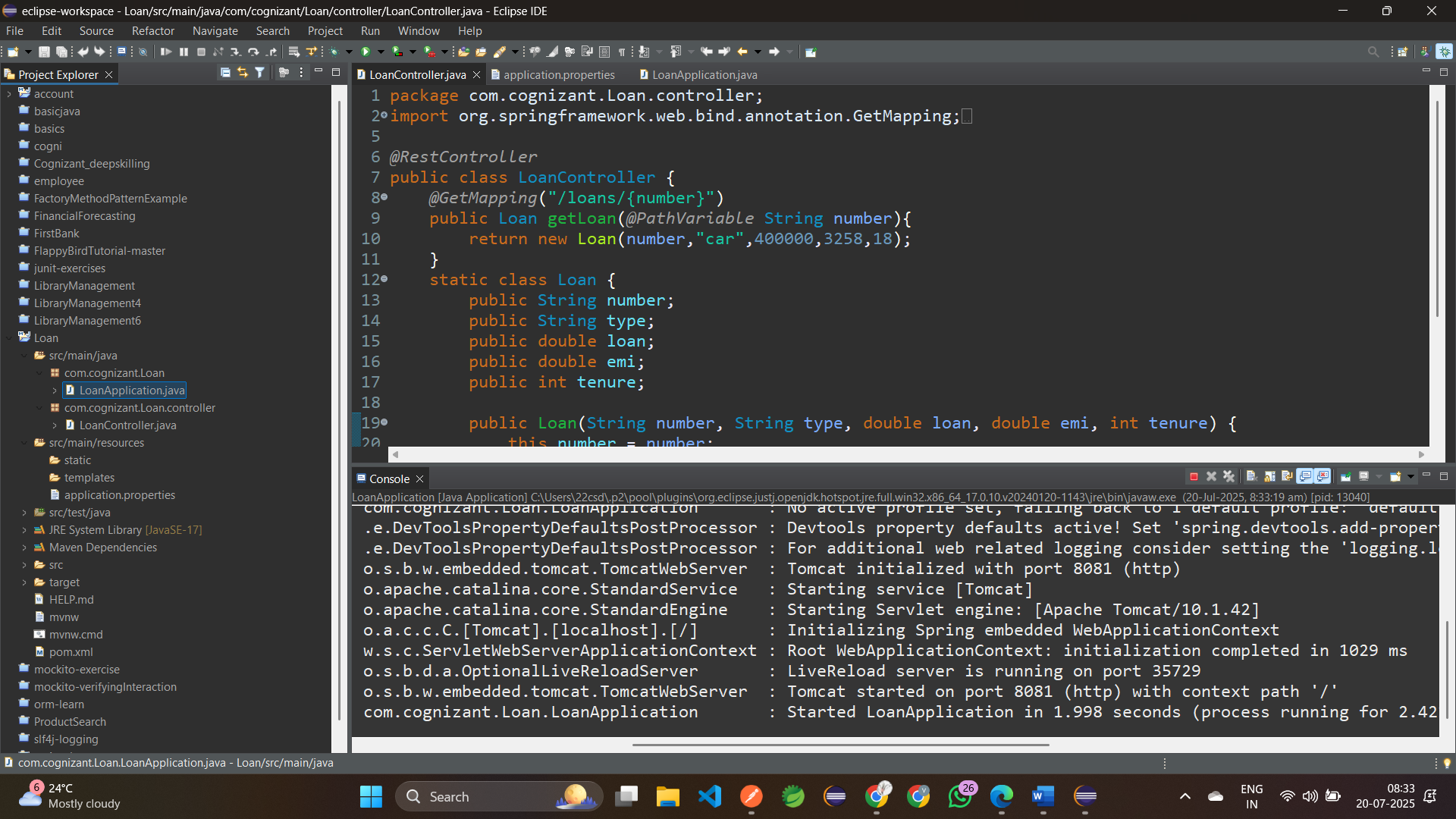


4. Change the Port to 8081 in **application.properties**:

spring.application.name=Loan

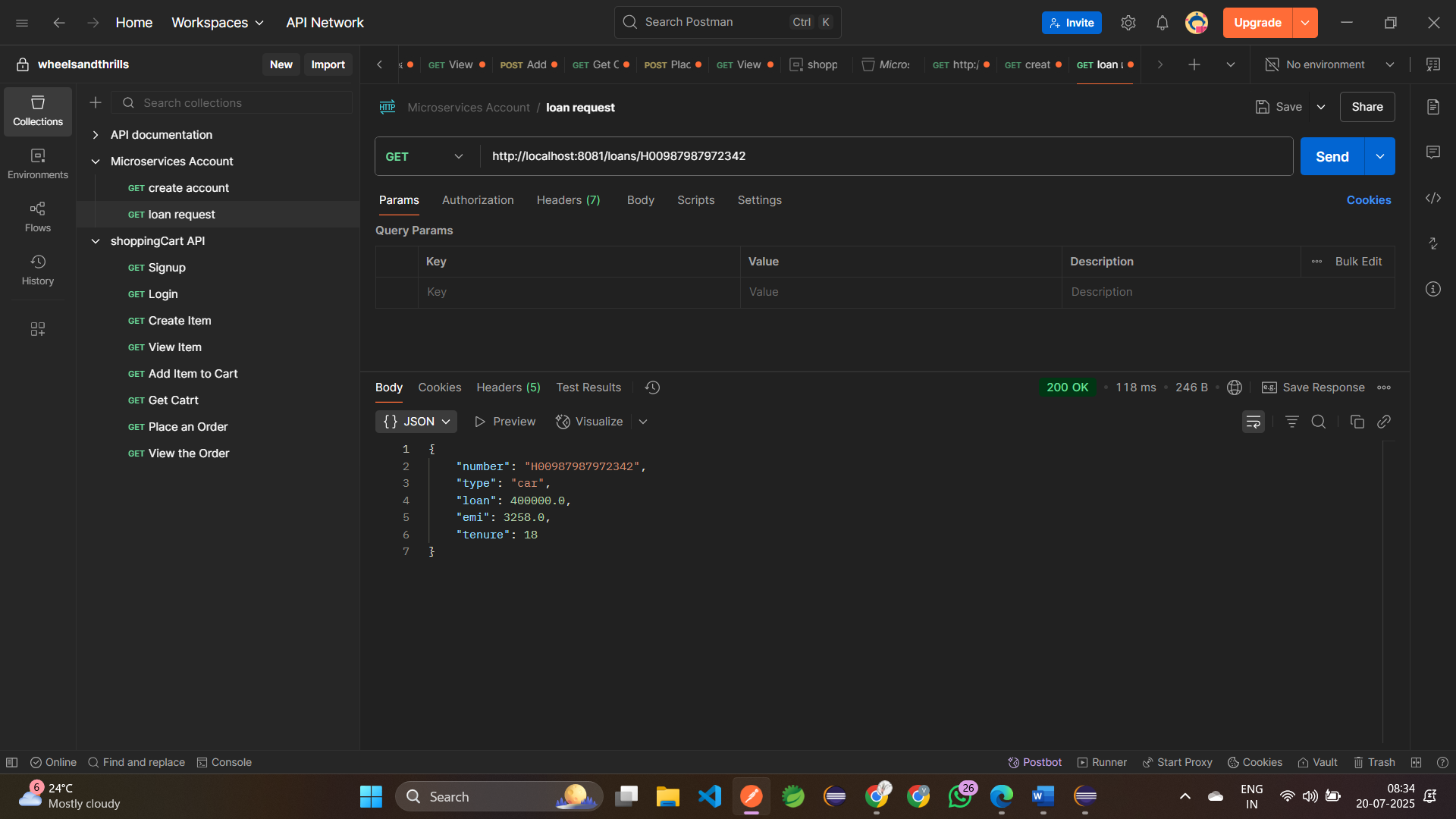
server.port=8081

5. Run the Application Again: **LoanApplication.java**



4. Testing:

i) Open postman  
 ii) Type the url: <http://localhost:8081/loans/H00987987972342>



• To switch between different consoles use the monitor icon within the console view.

