**2. Microservices with API gateway**

# 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.

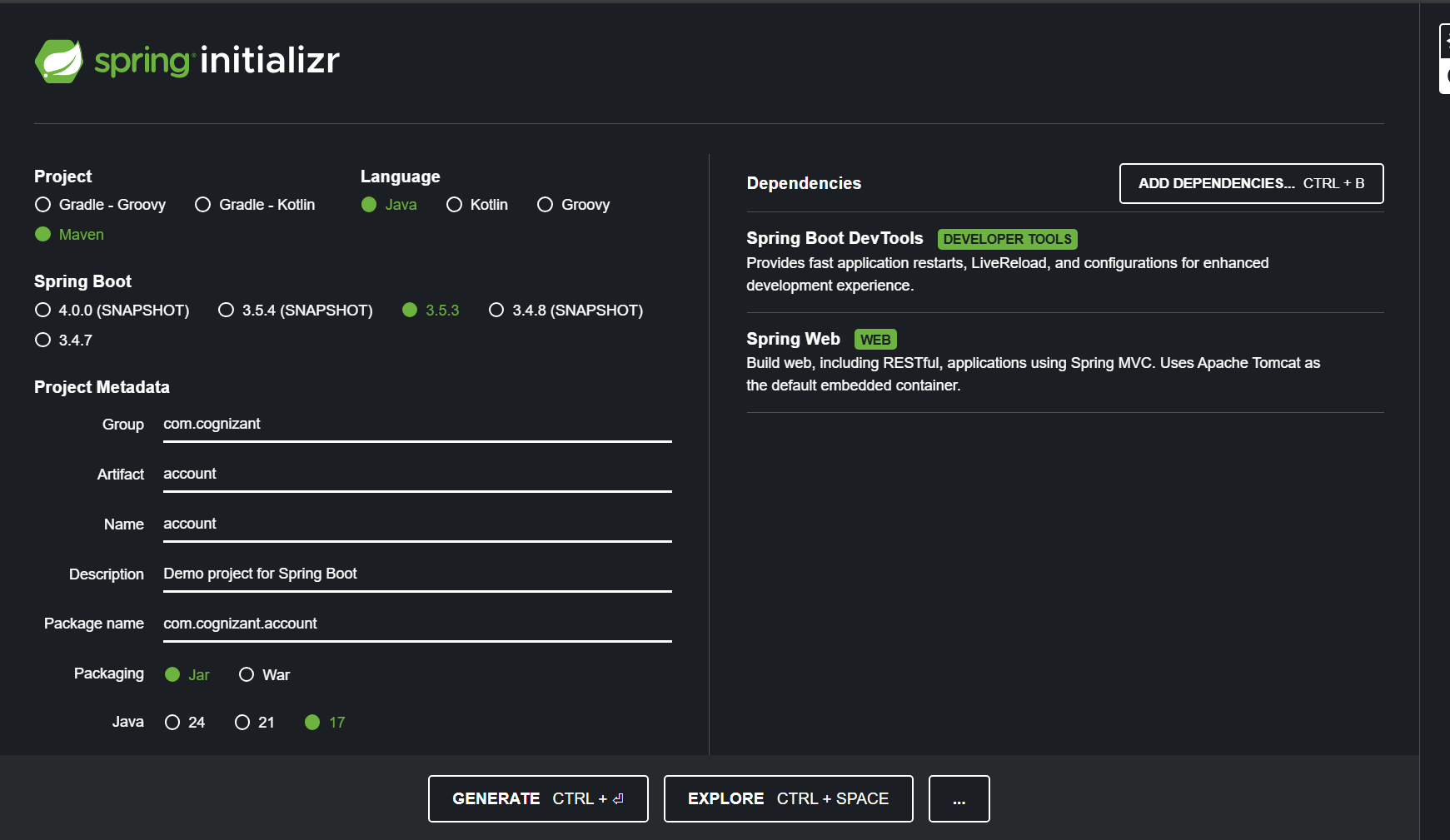
Each microservice will be a specific independent Spring RESTful Webservice maven project having it's own pom.xml. The only difference is that, instead of having both account and loan as a single application, it is split into two different applications. These webservices will be a simple service without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

Follow steps below to implement the two microservices:

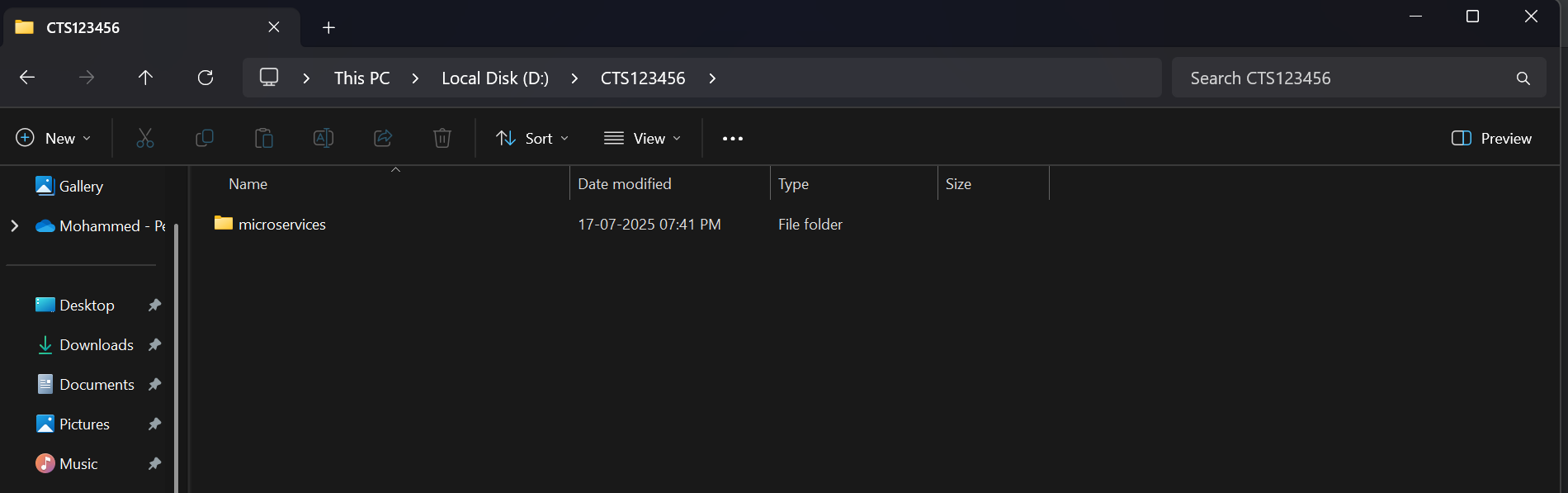
## Account Microservice

* 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. Refer specification below:
  + Method: GET
  + Endpoint: /accounts/{number}
  + Sample Response. Just a dummy response without any backend connectivity.

{ number: "00987987973432", type: "savings", balance: 234343 }

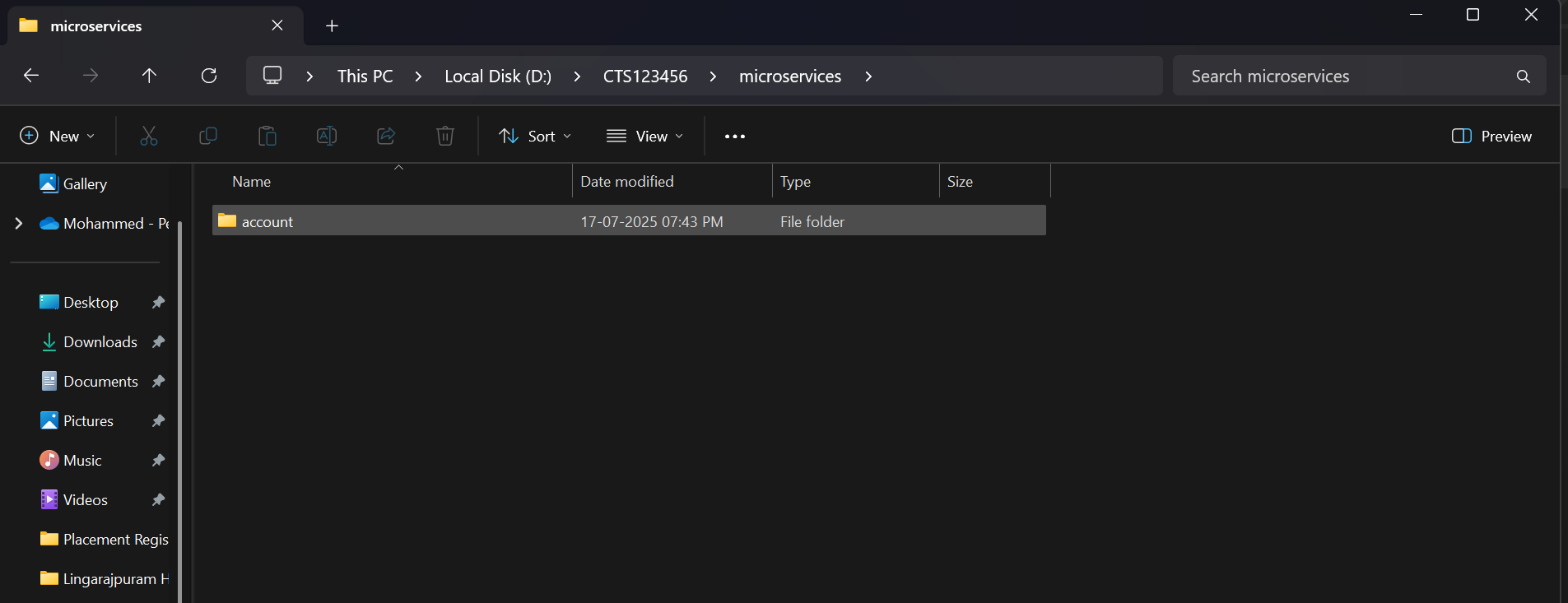
****

**Create a folder in D drive**

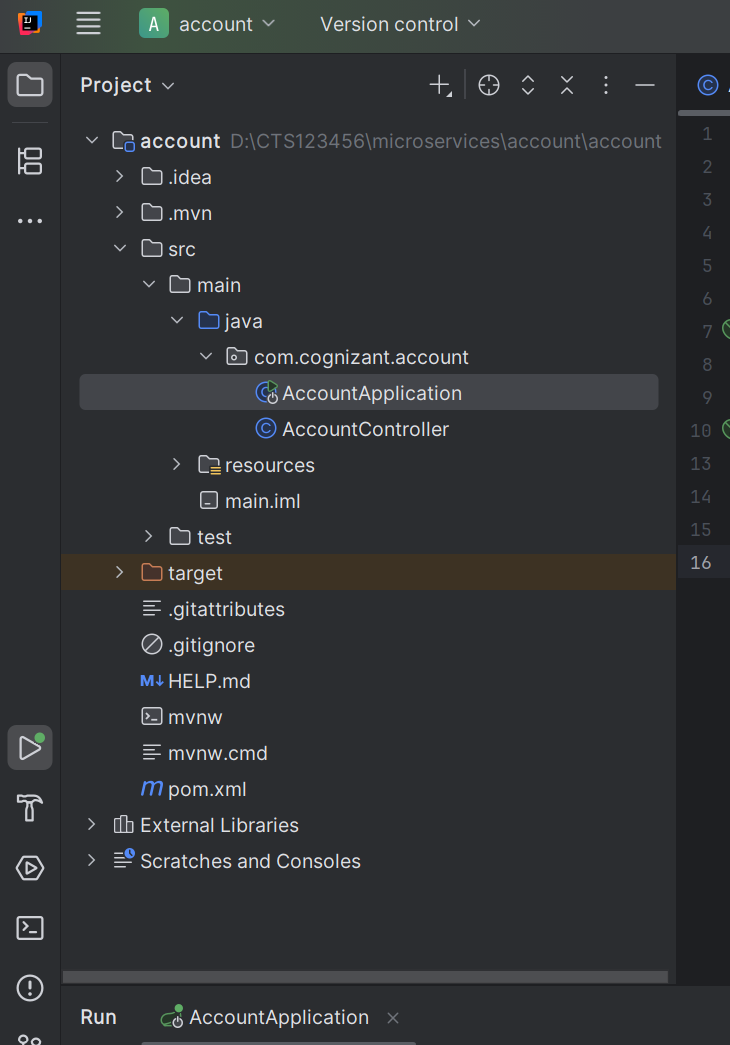
****

**Create a folder named microservices inside the created folder**

**Then, extract the spring initializr file and paste it inside the “microservices” folder**

****

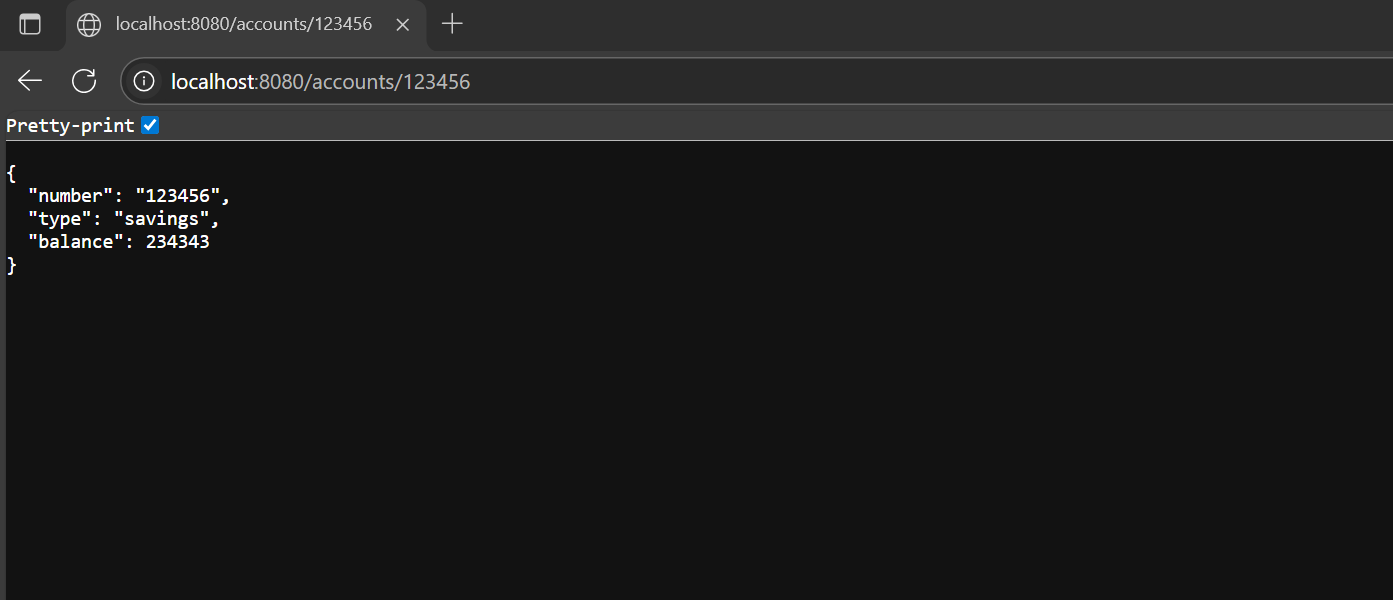
**Folder Structure:**

****

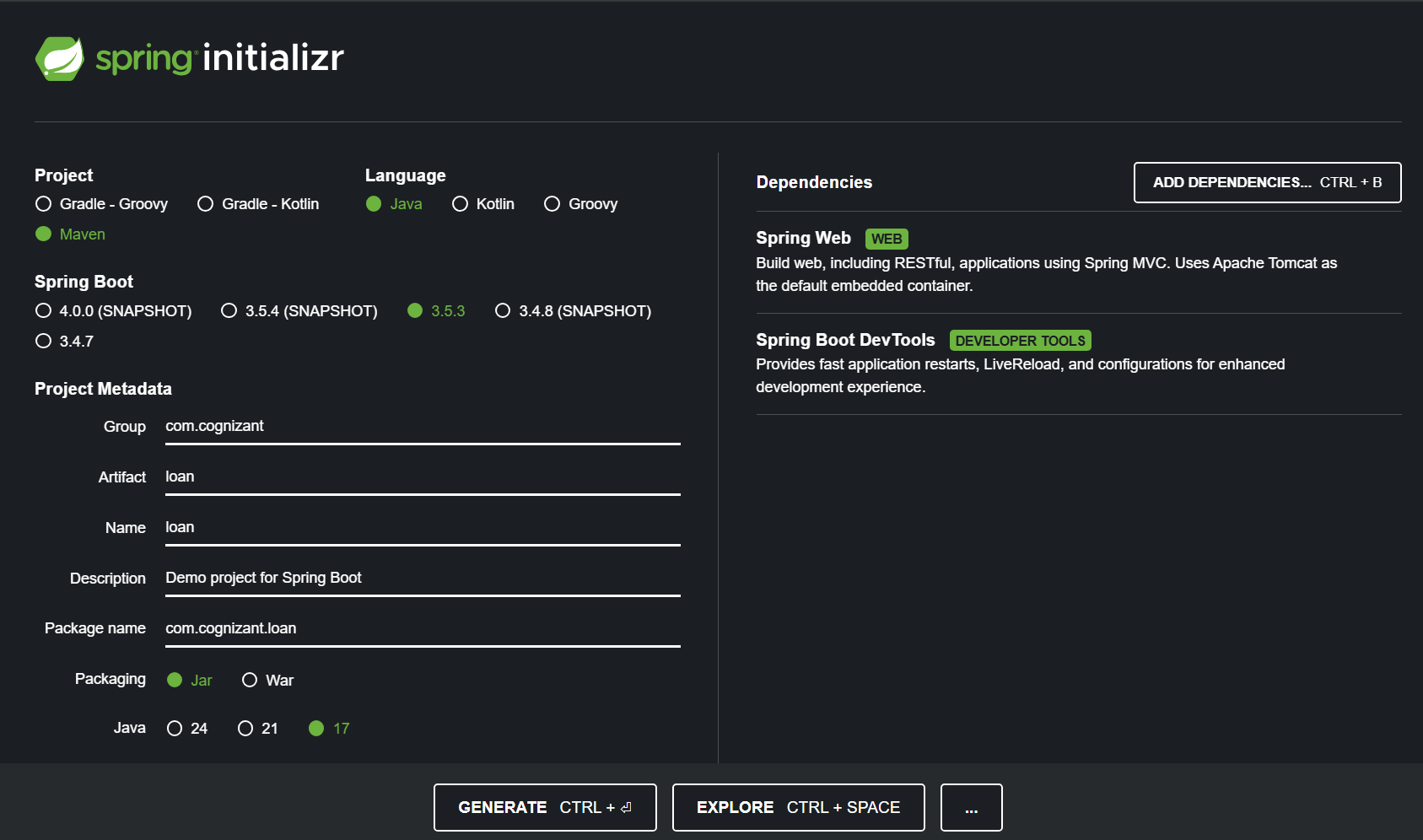
**AccountController.java**

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

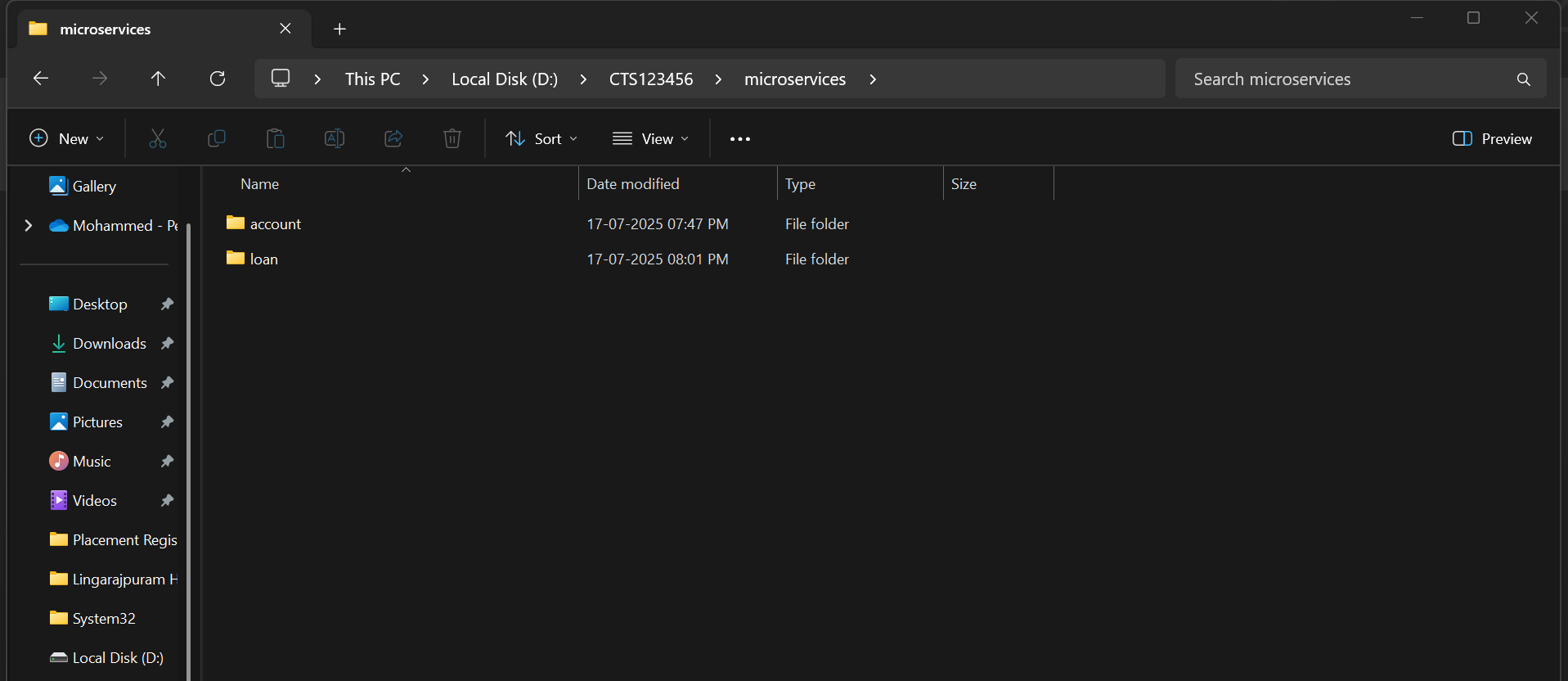
**Then, run AccountApplication.java and go to** [**localhost:8080/accounts/123456**](http://localhost:8080/accounts/123456)

****

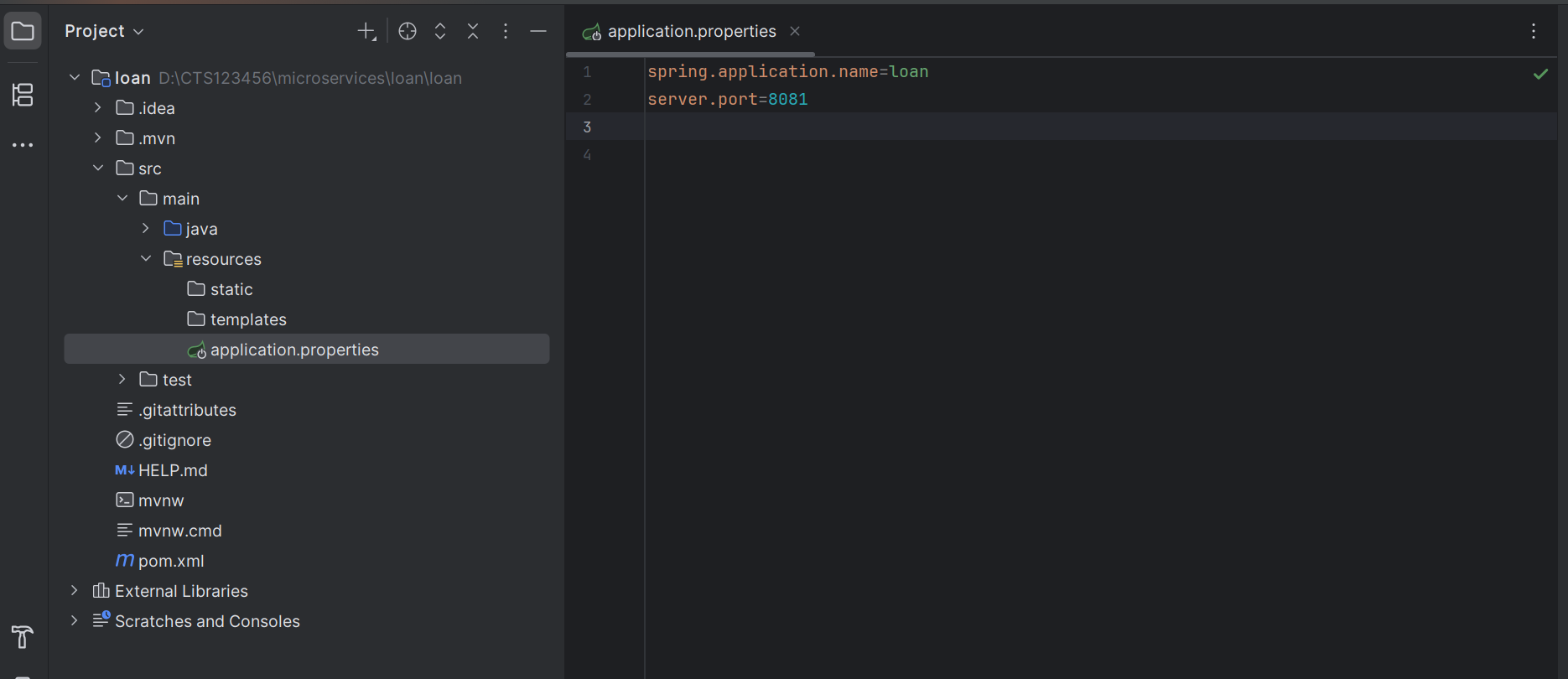
**Creating Loan microservice:**

****

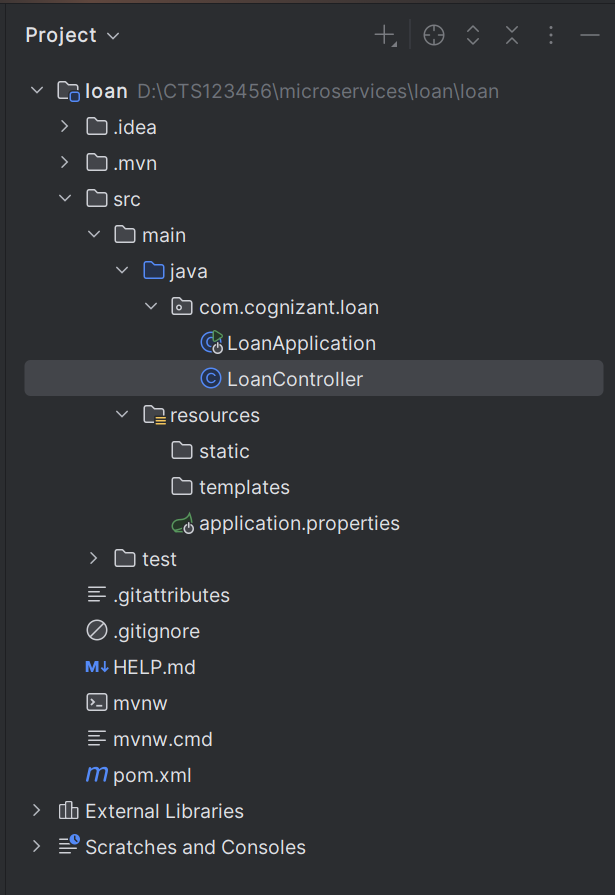
**Then move it to the same old folder,**

****

**Set custom port for the loans project by adding 🡪 server.port=8081**

****

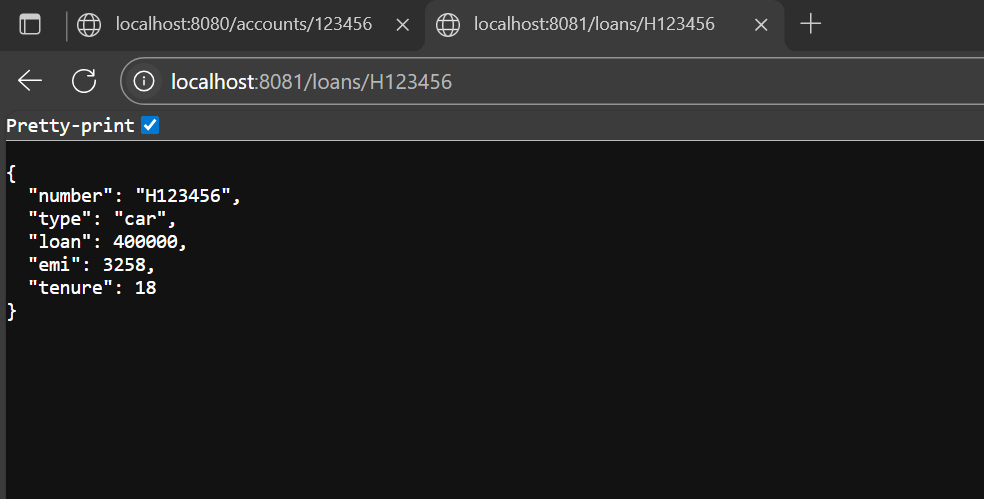
**Folder structure:**

****

**LoanController.java**

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

**Run LoanApplication.java and go to** [**http://localhost:8081/loans/H123456**](http://localhost:8081/loans/H123456)

****

I now have two microservices,

1. Account on port 8080
2. Loan on port 8081

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ThankYou**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_