**MICROSERVICES**

1. 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.

**Solution:**

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:

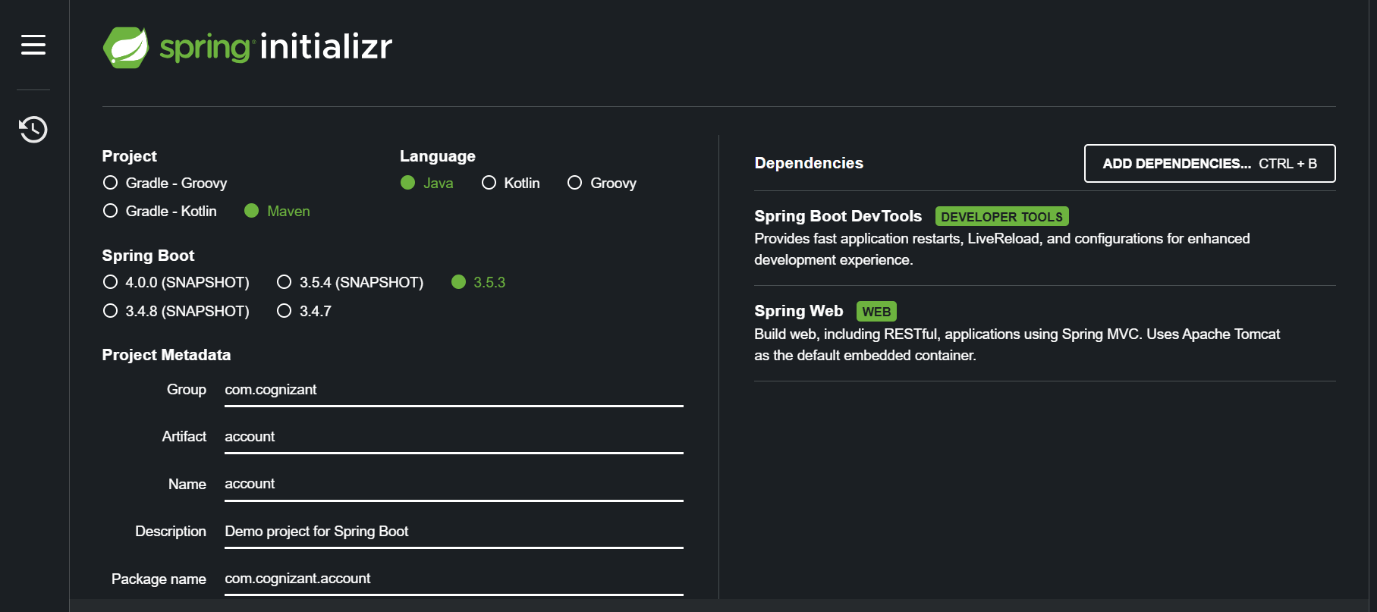
Group: com.cognizant

Artifact: account

* Select the following modules

Developer Tools > Spring Boot Dev Tools

Web > Spring Web



**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);

}

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; }

}

}

**AccountApplication.java**  
  
package com.cognizant.account;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class AccountApplication {

public static void main(String[] args) {

SpringApplication.run(AccountApplication.class, args);

}

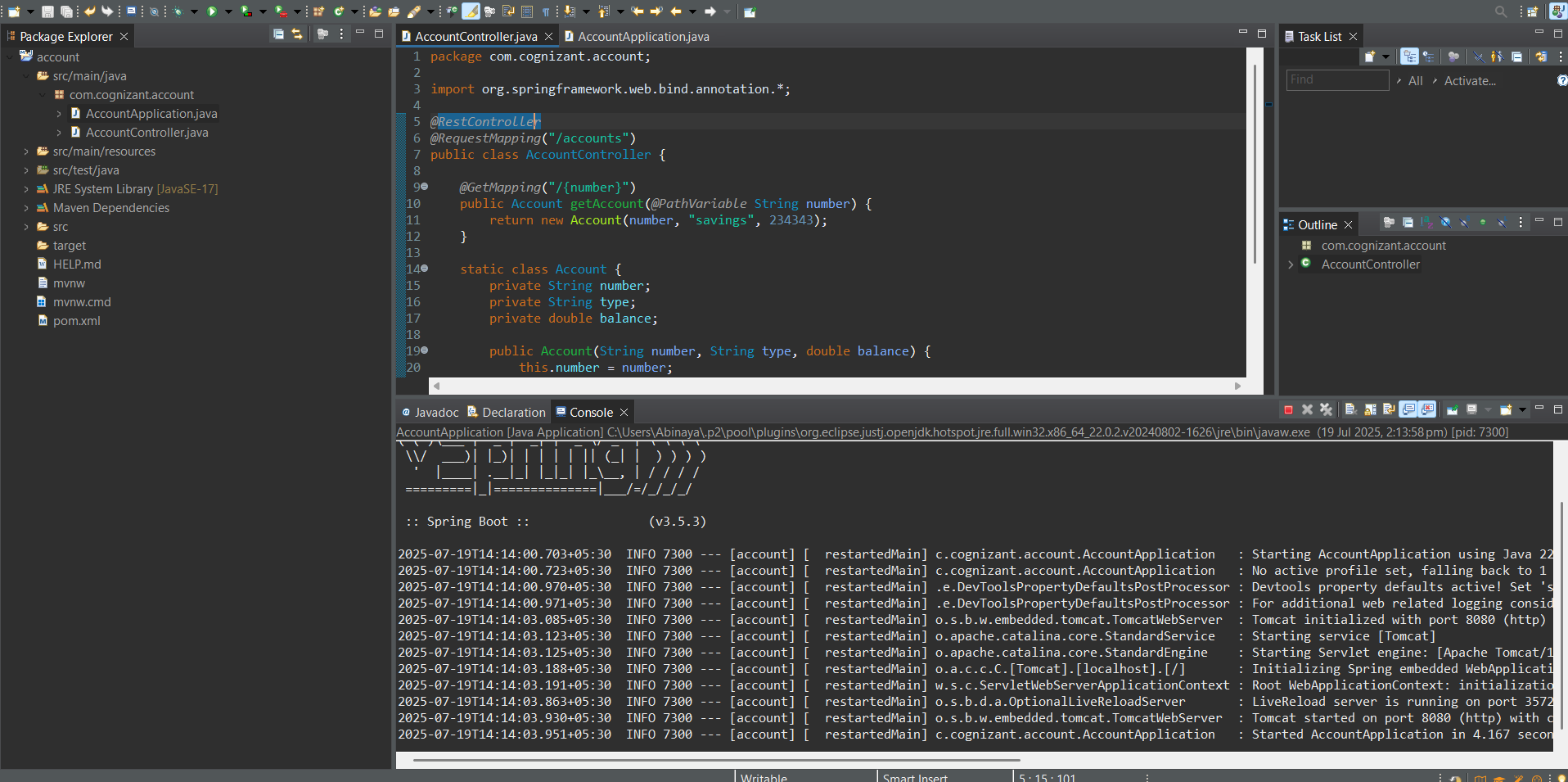
}

Import this project in Eclipse and implement a controller method for getting account details based on account number. Refer specification below:

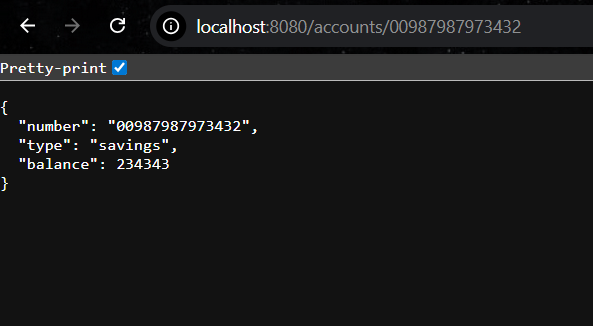
o Method: GET

o Endpoint: /accounts/{number}

o Sample Response.

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

**Output:**



**Loan Microservice**

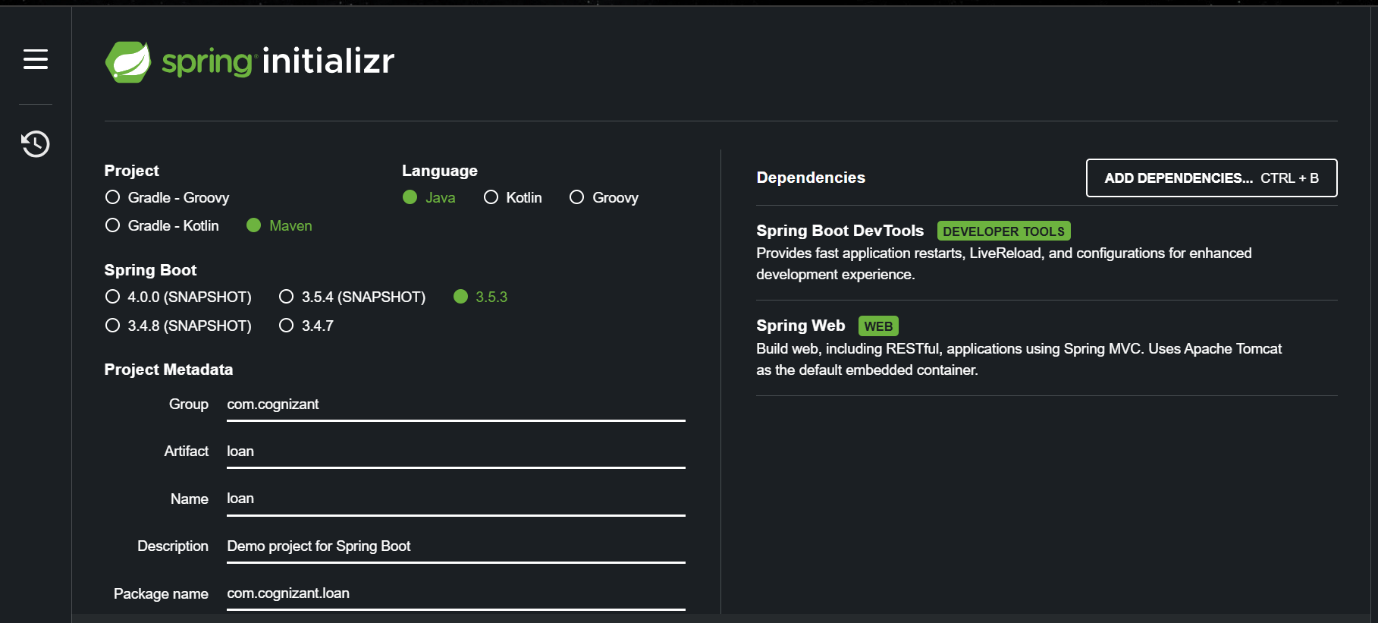
Follow similar steps specified for Account Microservice and implement a service API to get loan account details

o Method: GET

o Endpoint: /loans/{number}

o Sample Response.

{ number: "H00987987972342", type: "car", loan: 400000, emi: 3258, tenure: 18 }



**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);

}

static class Loan {

private String number;

private String type;

private int loan;

private int emi;

private int tenure;

public Loan(String number, String type, int loan, int 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 int getLoan() { return loan; }

public int getEmi() { return emi; }

public int getTenure() { return tenure; }

}

}

**LoanController.java**

package com.cognizant.loan;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class LoanApplication {

public static void main(String[] args) {

SpringApplication.run(LoanApplication.class, args);

}

}

**Output:**

