**WEEK 5 Hands\_on :**

**Creating Micro-Services for Account and Loan**

**Task Description: Creating Microservices for Account and Loan**

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.

**CODE AND OUTPUT :**

**1 . CREATE ACCOUNT MICROSERVICE :**

**Step 1 :** **Setting up springBoot Application** :

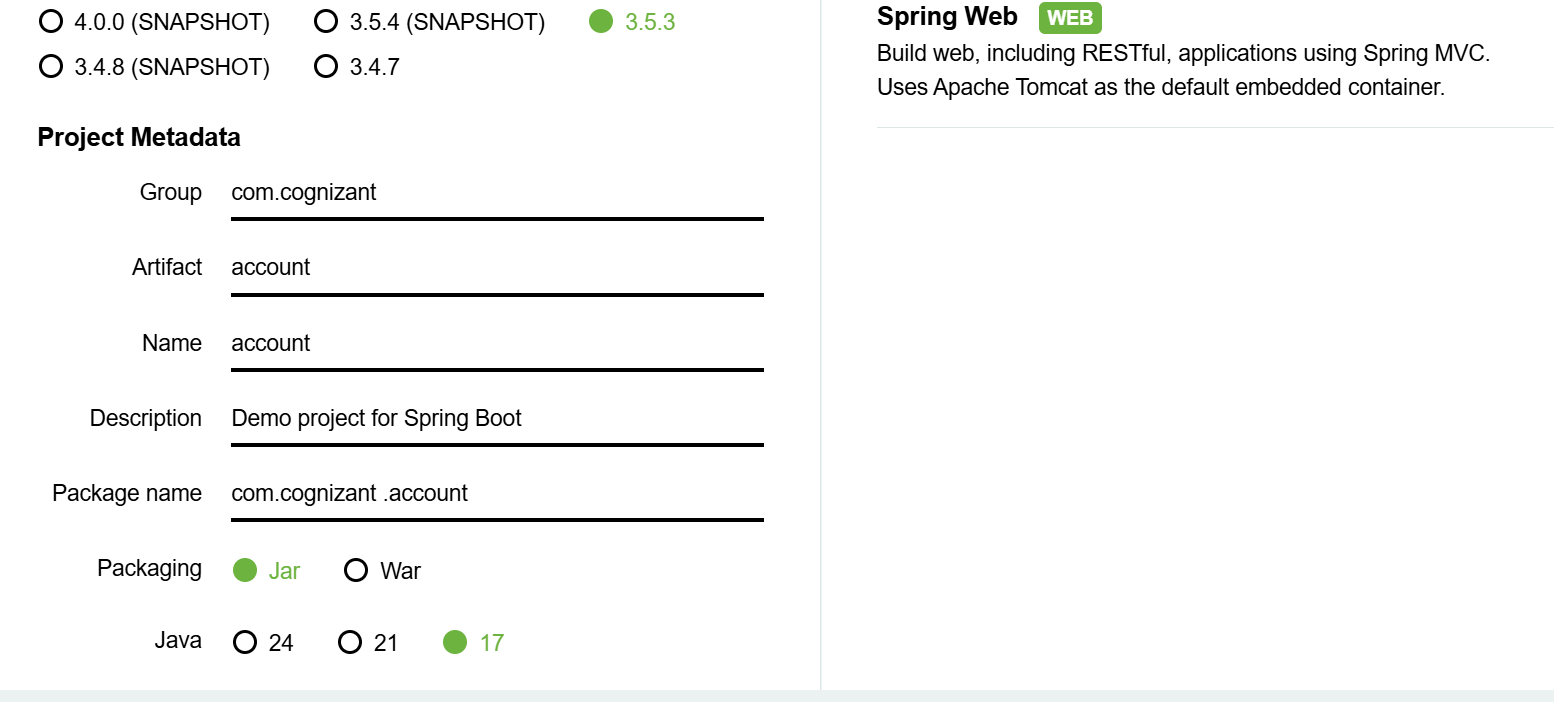
Open : <https://start.spring.io/>

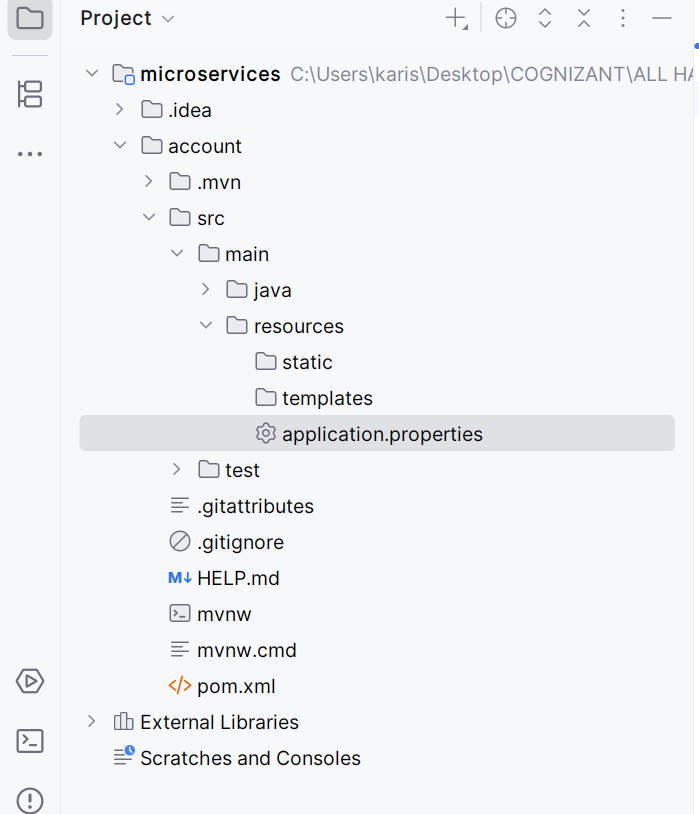
Next enter :

Group: com.cognizant

Artifact: account

After that click on generate , a zip file will downloaded.extract it it and open in Eclipse /Intellij Ide





**Step 2 . Application.properties :**

spring.application.name=account   
server.port=8083

**Step 3 . Account Application.java:**

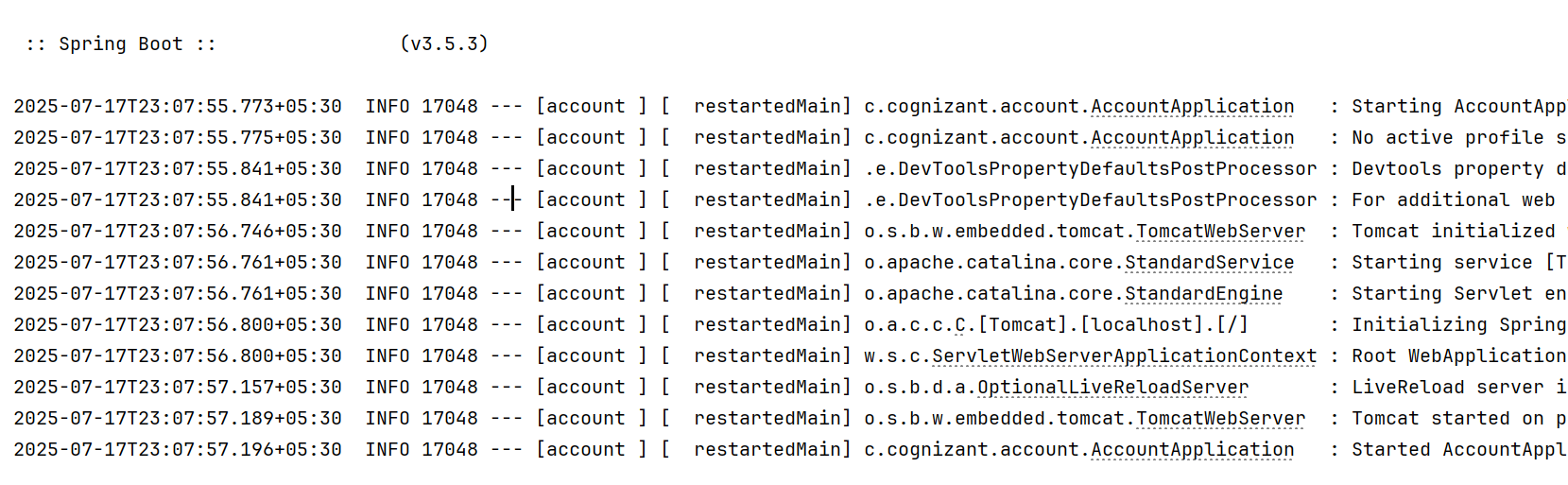
package com.cognizant.account;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class AccountApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(AccountApplication.class, args);  
 }  
}

**Step 4. Create a package controller :** package com.cognizant.account.controller;

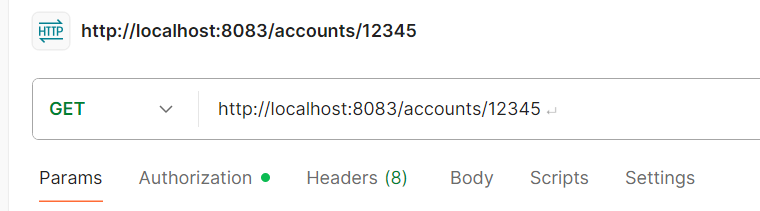
Inside that package create a **AccountController.java** class:

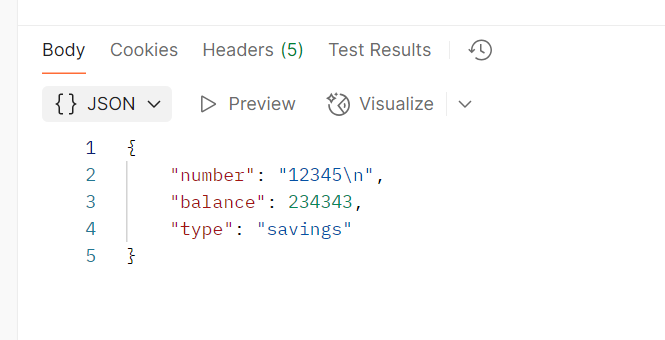
package com.cognizant.account.controller;  
  
import org.springframework.web.bind.annotation.\*;  
import java.util.\*;  
  
@RestController  
@RequestMapping("/accounts")  
public class AccountController {  
  
 @GetMapping("/{number}")  
 public Map<String, Object> getAccount(@PathVariable String number) {  
 Map<String, Object> account = new HashMap<>();  
 account.put("number", number);  
 account.put("type", "savings");  
 account.put("balance", 234343);  
 return account;  
 }  
}

**Step 5. Run the Application**



**Step 6. Test the connection in postman :**





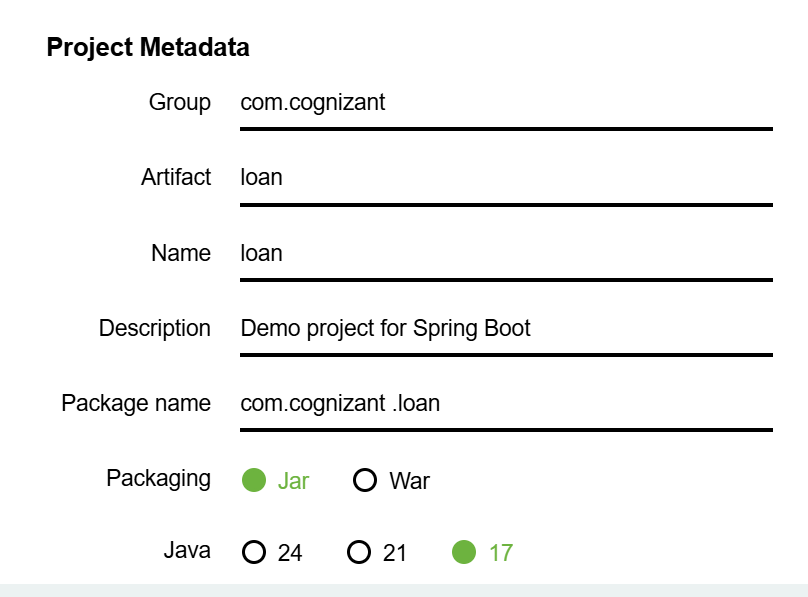
**2 . CREATE LOAN MICROSERVICE :**

**Step 1 . Configuring and creating springBoot application for loan microservice :**

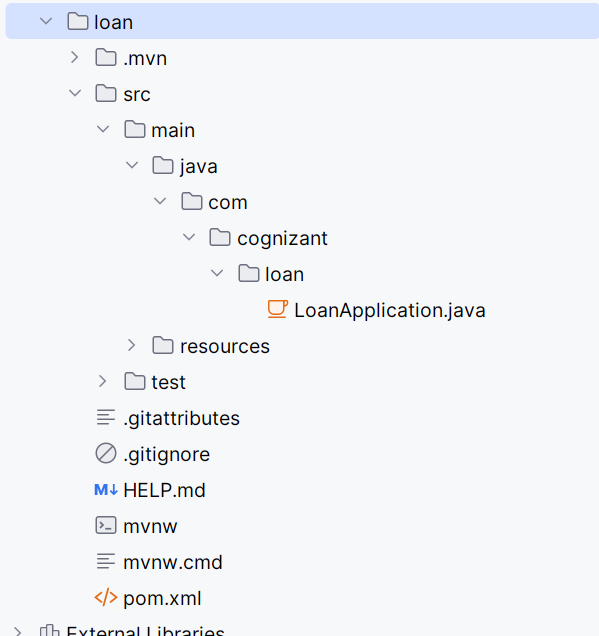
Open [Spring Initializr](https://start.spring.io/)

Fill details:

* Group: com.cognizant
* Artifact: loan
* Dependencies:
  + Spring Web
  + Spring Boot DevTools



**Download and extract and open :**

****

**Step 2 . Application.properties :**

**Configure port :**

spring.application.name=loan  
server.port=8084

**Step 3 . LoanApplication.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);

} }

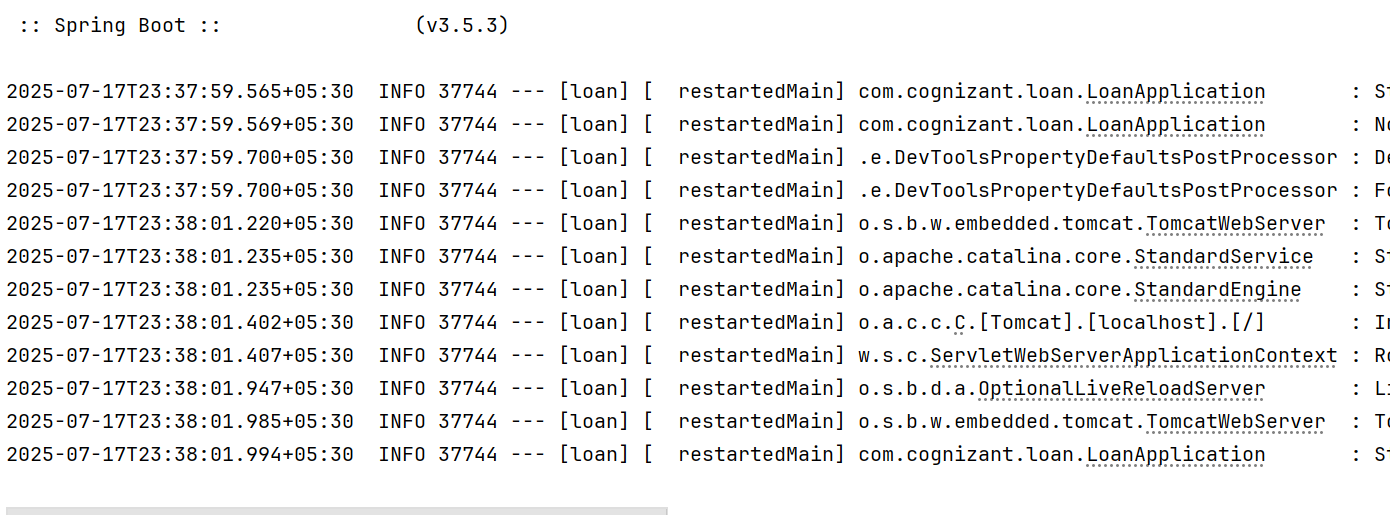
**Step 4. Create a package controller and create a controller class :**

**Create LoanController.java in src/main/java/com/cognizant/loan/controller**

package com.cognizant.loan.controller;  
  
import org.springframework.web.bind.annotation.\*;  
import java.util.\*;  
  
@RestController  
@RequestMapping("/loans")  
public class LoanController {  
  
 @GetMapping("/{number}")  
 public Map<String, Object> getLoan(@PathVariable String number) {  
 Map<String, Object> loan = new HashMap<>();  
 loan.put("number", number);  
 loan.put("type", "car");  
 loan.put("loan", 400000);  
 loan.put("emi", 3258);  
 loan.put("tenure", 18);  
 return loan;  
 }  
}

**Step 5. Run the Application :**

**Run LoanApplication.java (main class)**

****

**Step 6. Test the connection in postman :**

