WEEK – 5

Creating Microservices for account and loan

Nipuna

Superset id: 6432842

**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 }

Solution:

1)AccountApplication.java:

**package** com.cognizant.account;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

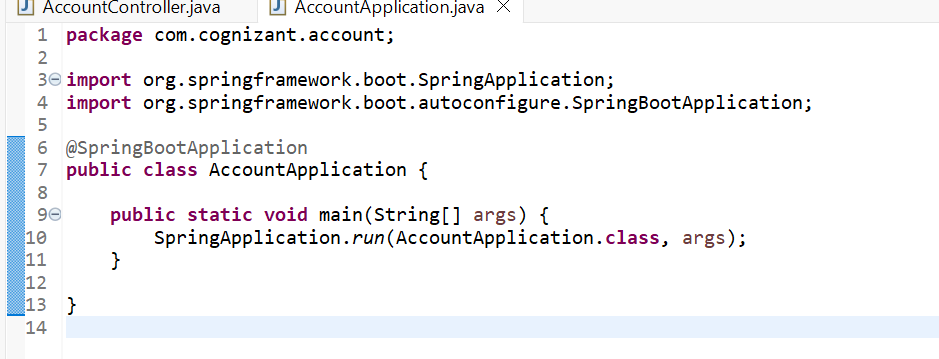
**public** **class** AccountApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(AccountApplication.**class**, args);

}

}



2)AccountController.java:

**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;

**import** java.util.Map;

@RestController

**public** **class** AccountController {

@GetMapping("/accounts/{number}")

**public** Map<String, String> getAccountByNumber(@PathVariable String number) {

**return** Map.*of*(

"accountNumber", number,

"accountHolder", "John Doe",

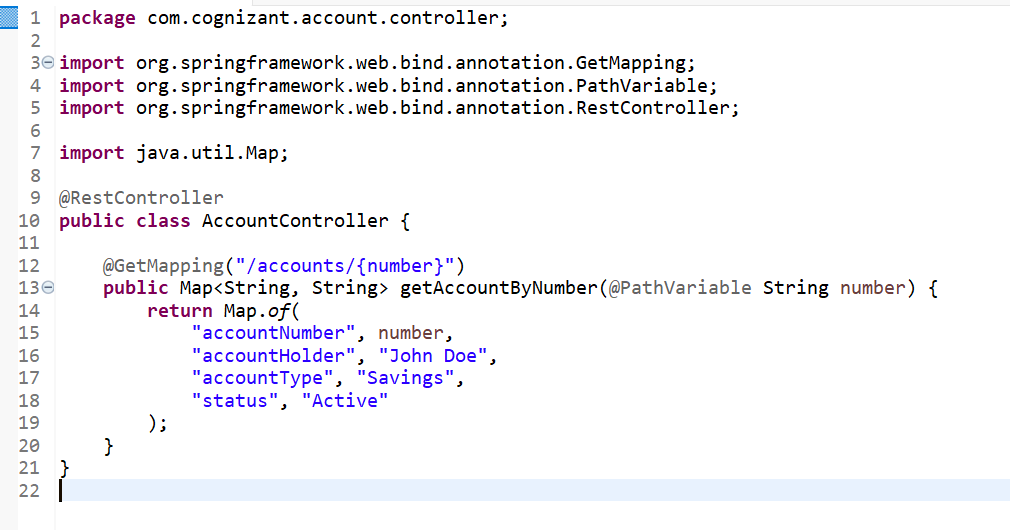
"accountType", "Savings",

"status", "Active"

);

}

}



Output:

