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:

o Group: com.cognizant

o Artifact: account

□ Select the following modules

o Developer Tools > Spring Boot DevTools

o 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:

o Method: GET

o Endpoint: /accounts/{number}

o Sample Response. Just a dummy response without any backend connectivity.

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

□ Launch by running the application class and test the service in browser

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. Just a dummy response without any backend connectivity.

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

□ Launching this application by having account service already running

□ This launch will fail with error that the bind address is already in use

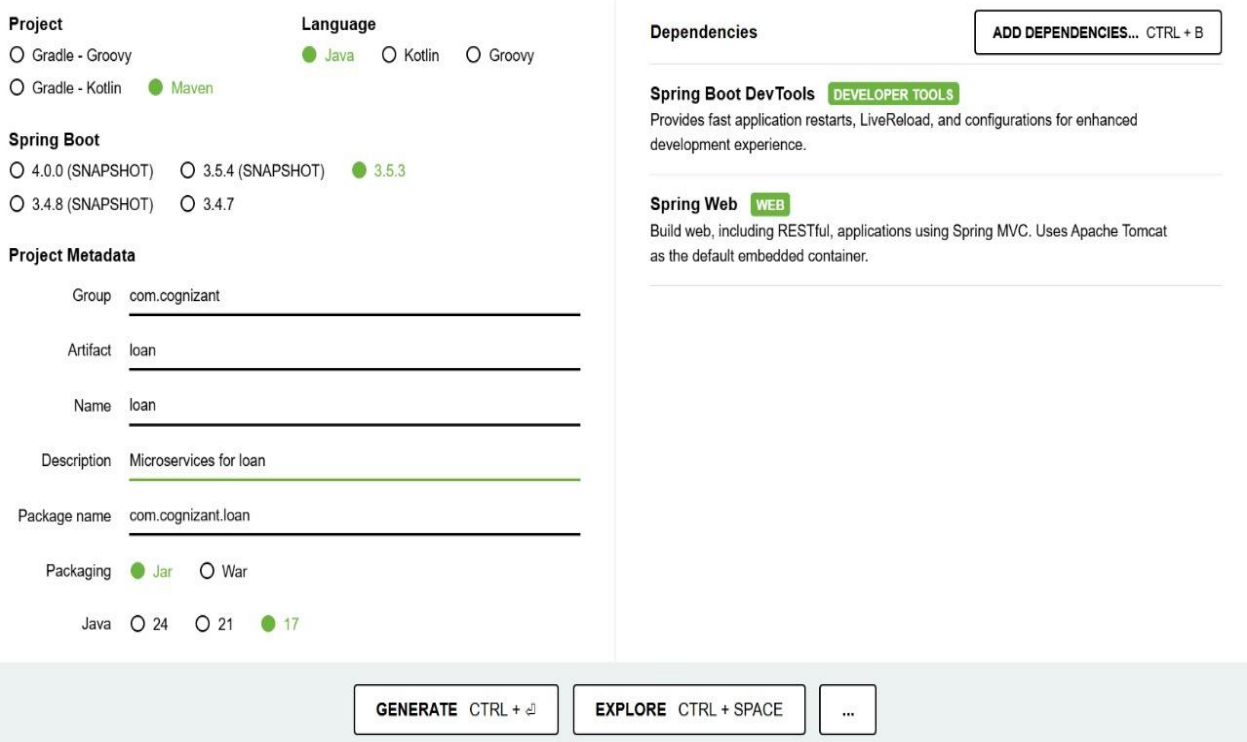
⍰ The reason is that each one of the service is launched with default port number as 8080. Account service is already using this port and it is not available for loan service.

⍰ Include "server.port" property with value 8081 and try launching the application

⍰ Test the service with 8081 port

Now we have two microservices running on different ports.

NOTE: The console window of Eclipse will have both the service console running. To switch between different consoles use the monitor icon within the console view



Project

- ○ Gradle - Groovy
- ○ Gradle - Kotlin
- ● Maven

Language

- ● Java
- ○ Kotlin
- ○ Groovy

Spring Boot

- ○ 4.0.0 (SNAPSHOT)
- ○ 3.5.4 (SNAPSHOT)
- ● 3.5.3
- ○ 3.4.8 (SNAPSHOT)
- ○ 3.4.7

Project Metadata

Group	com.cognizant
Artifact	account
Name	account
Description	Microservices for account
Package name	com.cognizant.account
Packaging	● Jar ○ War
Java	○ 24 ○ 21 ● 17

Dependencies — ADD DEPENDENCIES... CTRL + B

Spring Boot DevTools DEVELOPER TOOLS
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

GENERATE CTRL + ⏎ | EXPLORE CTRL + SPACE | ...

Recent download history

loan.zip
14.7 KB • 1 minute ago

account.zip
14.9 KB • 3 minutes ago

Full download history

Account Microservice

AccountController.java

```
package com.cognizant.account.controller;


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);
   }
}
```

Account.java (Model Class)

```
package com.cognizant.account.controller;
public 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;
    }
}
```

Launch and Test:

Open browser or Postman:

http://localhost:8080/accounts/00987987973432

output:


```
"number": "00987987973432",
 "type": "savings",
"balance": 234343
```


Loan Microservice

LoanController.java

```
package com.cognizant.loan.controller;
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);
    }
}
```

Loan.java (Model Class)

```
package com.cognizant.loan.controller;
public class Loan {
    private String number;
    private String type;
    private double loan;
    private int emi;
    private int tenure;
 public Loan(String number, String type, double 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 double getLoan() {
        return loan;
    }
public int getEmi() {
        return emi;
    }
  public int getTenure() {
        return tenure;
    }
}
```

Launch and Test

Open browser or Postman:

http://localhost:8081/loans/H00987987972342

Output:


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


KANMANI MURUGHAIYAN

SUPERSET ID: 6407636