

Week – 5

Skill: Microservices with Spring Boot 3 and Spring Cloud

Exercise: Creating Microservices for account and loan

- account-service

```
Account.java x AccountController.java application.properties Ac
1 package com.bank.account_service.model;
2 import jakarta.persistence.*;
3 @Entity 6 usages
4 public class Account {
5     @Id 2 usages
6     @GeneratedValue(strategy = GenerationType.IDENTITY)
7     private Long id;
8     private String accountHoldername; 2 usages
9     private String accountType; 2 usages
10    private Double balance; 2 usages
11    public Long getId() { no usages
12        return id; }
13    public void setId(Long id) { no usages
14        this.id = id; }
15    public String getAccountHoldername() { no usages
16        return accountHoldername; }
17    public void setAccountHoldername(String accountHoldername) {
18        this.accountHoldername = accountHoldername; }
19    public String getAccountType() { no usages
20        return accountType;
21    }
22    public void setAccountType(String accountType) { no usages
23        this.accountType = accountType; }
24    public Double getBalance() { no usages
25        return balance; }
26    public void setBalance(Double balance) { no usages
27        this.balance = balance;
28    }
29 }
```

```
account-service C:\Users\shalu\Downloads
D:\idea
account-service
  .mvn
  src
  main
    java
      com.bank.account_service
        controller
          AccountController
        model
        repository
          AccountRepository
          AccountServiceApplication
      resources
        static
        templates
        application.properties
    test
1 package com.bank.account_service.controller;
2 import com.bank.account_service.model.Account;
3 import com.bank.account_service.repository.AccountRepository;
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.web.bind.annotation.*;
6 import java.util.List;
7 @RestController no usages
8 @RequestMapping("/accounts")
9 public class AccountController {
10    @Autowired 2 usages
11    private AccountRepository accountRepository;
12
13    @PostMapping no usages
14    public Account createAccount(@RequestBody Account account) {
15        return accountRepository.save(account);
16    }
17    @GetMapping no usages
18    public List<Account> getAllAccounts() {
19        return accountRepository.findAll();
20    }
21 }
```

```
Account.java AccountController.java application.properties AccountServiceApplicat
1 package com.bank.account_service.repository;
2 import com.bank.account_service.model.Account;
3 import org.springframework.data.jpa.repository.JpaRepository;
4 public interface AccountRepository extends JpaRepository<Account, Long> { 2 usages
5 }
6 }
```

```
Account.java AccountController.java application.properties x
1 server.port=8081
2 spring.application.name=account-service
3 spring.datasource.url=jdbc:h2:mem:accountdb
4 spring.datasource.driverClassName=org.h2.Driver
5 spring.datasource.username=sa
6 spring.datasource.password=
7 spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
8 spring.jpa.hibernate.ddl-auto=update
9 spring.h2.console.enabled=true
10
```

POST http://localhost:8081/accounts Send

Params Auth Headers (8) Body Scripts Settings Cookies

raw JSON Beautify

```
1 {
2   "accountHoldername": "Alice",
3   "accountType": "Savings",
4   "balance": 15000
5 }
6
```

Body 200 OK • 1.17 s • 242 B

{ JSON Preview Visualize

```
1 {
2   "id": 1,
3   "accountHoldername": "Alice",
4   "accountType": "Savings",
5   "balance": 15000.0
6 }
```

- *Loan-service*

The screenshot shows an IDE with a project named 'loan-service'. The project structure is visible on the left, showing a 'src/main/java/com.bank.loan_service/controller' directory containing 'LoanController.java'. The 'LoanController.java' file is open in the editor, showing the following code:

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import java.util.List;

@RestController
@RequestMapping("/loans")
public class LoanController {
    @Autowired
    private LoanRepository loanRepository;

    @PostMapping
    public Loan createLoan(@RequestBody Loan loan) {
        return loanRepository.save(loan);
    }

    @GetMapping
    public List<Loan> getAllLoans() {
        return loanRepository.findAll();
    }
}

```

The screenshot shows the 'LoanRepository.java' file in the IDE. The code defines a 'Loan' entity with the following attributes and methods:

```

package com.bank.loan_service.model;
import jakarta.persistence.*;

@Entity
public class Loan {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String loanType;
    private Double loanAmount;
    private Long accountId;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getLoanType() {
        return loanType;
    }

    public void setLoanType(String loanType) {
        this.loanType = loanType;
    }

    public Double getLoanAmount() {
        return loanAmount;
    }

    public void setLoanAmount(Double loanAmount) {
        this.loanAmount = loanAmount;
    }

    public Long getAccountId() {
        return accountId;
    }

    public void setAccountId(Long accountId) {
        this.accountId = accountId;
    }
}

```

The screenshot shows the 'LoanServiceApplication.java' file in the IDE. The code defines a Spring Boot application with the following structure:

```

package com.bank.loan_service;

import ...

@SpringBootApplication
public class LoanServiceApplication {
    public static void main(String[] args) {
        SpringApplication.run(LoanServiceApplication.class, args);
    }
}

```

The screenshot shows the 'application.properties' file in the IDE. The configuration properties are as follows:

```

spring.application.name=loan-service
server.port=8082
spring.datasource.url=jdbc:h2:mem:loandb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.h2.console.enabled=true

```

HTTP

http://localhost:8082/loans

Save

Share

POST

http://localhost:8082/loans

Send

Params

Auth

Headers (8)

Body

Scripts

Settings

Cookies

Beautiful

raw

JSON

```
1 {
2   "loanType": "Home Loan",
3   "loanAmount": 500000,
4   "accountId": 1
5 }
6
```

Body

200 OK

1.63 s

231 B

Visualize

Preview

JSON

```
1 {
2   "id": 1,
3   "loanType": "Home Loan",
4   "loanAmount": 500000.0,
5   "accountId": 1
6 }
```