**Creating Microservices for account and loan:**

**ACCOUNT:**

**Project Structure:**

account-service/

├── src/

│ └── main/

│ └── java/

│ └── com/

│ └── cognizant/

│ └── account/

│ ├── AccountApplication.java

│ ├── model/

│ │ └── Account.java

│ ├── repository/

│ │ └── AccountRepository.java

│ └── controller/

│ └── AccountController.java

└── resources/

├── application.properties

**application.properties:**

spring.datasource.url=jdbc:mysql://localhost:3306/bank

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

**Account.java:**

package com.cognizant.account.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

@Entity

public class Account {

@Id

private Long accountId;

private String accountHolderName;

private Double balance;

// Getters and Setters

}

**AccountRepository.java**

package com.cognizant.account.repository;

import com.cognizant.account.model.Account;

import org.springframework.data.jpa.repository.JpaRepository;

public interface AccountRepository extends JpaRepository<Account, Long> {

}

**AccountController.java:**

package com.cognizant.account.controller;

import com.cognizant.account.model.Account;

import com.cognizant.account.repository.AccountRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/accounts")

public class AccountController {

@Autowired

private AccountRepository accountRepository;

@GetMapping

public List<Account> getAllAccounts() {

return accountRepository.findAll();

}

@GetMapping("/{id}")

public Account getAccountById(@PathVariable Long id) {

return accountRepository.findById(id).orElse(null);

}

}

**pom.xml:**

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.cognizant</groupId>

<artifactId>account</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>account</name>

<description>Account Microservice</description>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.7.10</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<properties>

<java.version>11</java.version>

</properties>

<dependencies>

<!-- Spring Boot Web -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<!-- Spring Data JPA -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<!-- MySQL Connector -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

</dependency>

<!-- Spring Boot Test -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<!-- Spring Boot Maven Plugin -->

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**OUTPUT:**

A computer screen with a black screen

AI-generated content may be incorrect.

**LOAN:**

loan-service/

├── src/

│ └── main/

│ ├── java/

│ │ └── com/example/loan/

│ │ ├── controller/

│ │ │ └── LoanController.java

│ │ ├── model/

│ │ │ └── Loan.java

│ │ └── LoanServiceApplication.java

│ └── resources/

│ ├── application.properties

├── pom.xml

**LoanServiceApplication.java – Main Class:**

package com.example.loan;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

/\*\*

\* Entry point of the Loan Service Spring Boot application.

\*/

@SpringBootApplication

public class LoanServiceApplication {

public static void main(String[] args) {

SpringApplication.run(LoanServiceApplication.class, args);

}

}

**Loan.java – Model Class:**

package com.example.loan.model;

/\*\*

\* Loan model class representing loan details.

\*/

public class Loan {

private String number;

private String type;

private int loan;

private int emi;

private int tenure;

// Constructor

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;

}

// Getters and Setters

public String getNumber() {

return number;

}

public void setNumber(String number) {

this.number = number;

}

public String getType() {

return type;

}

public void setType(String type) {

this.type = type;

}

public int getLoan() {

return loan;

}

public void setLoan(int loan) {

this.loan = loan;

}

public int getEmi() {

return emi;

}

public void setEmi(int emi) {

this.emi = emi;

}

public int getTenure() {

return tenure;

}

public void setTenure(int tenure) {

this.tenure = tenure;

}

}

**LoanController.java – REST Controller:**

package com.example.loan.controller;

import com.example.loan.model.Loan;

import org.springframework.web.bind.annotation.\*;

/\*\*

\* REST controller that exposes loan-related endpoints.

\*/

@RestController

@RequestMapping("/loans")

public class LoanController {

/\*\*

\* Returns a dummy loan object for the given loan number.

\*

\* @param number the loan number

\* @return Loan details

\*/

@GetMapping("/{number}")

public Loan getLoanDetails(@PathVariable String number) {

// Returning mock loan data

return new Loan(number, "car", 400000, 3258, 18);

}

}

**pom.xml – Maven Dependencies:**

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>loan-service</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>loan-service</name>

<description>Loan Service Spring Boot Application</description>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.2.2</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<!-- Spring Boot Web Starter -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<!-- Optional: DevTools for auto reload -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<!-- Testing dependencies -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<!-- Spring Boot Maven Plugin -->

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**application.properties – Configuration:**

# Server will start on port 8081

server.port=8081

**OUTPUT:**

A computer screen with a black screen

AI-generated content may be incorrect.