Step – 1

Create basic SB microservice project

<groupId>com.eazybytes</groupId>

<artifactId>accounts</artifactId>

package com.eazybytes.accounts;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.autoconfigure.domain.EntityScan;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.ComponentScans;

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

@SpringBootApplication

@ComponentScans({@ComponentScan("com.eazybytes.accounts.controller")})

@EnableJpaRepositories("com.eazybytes.accounts.repository")

@EntityScan("com.eazybytes.accounts.model")

public class AccountsApplication {

public static void main(String[] args) {

SpringApplication.run(AccountsApplication.class, args);

}

}

accounts\src\main\resources\application.propertiesspring.application.name=accounts

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driver-class-name=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.h2.console.enabled=true

server.port=8080

accounts\src\main\resources\data.sql

INSERT INTO customer (name,email,mobile\_number,create\_dt)

VALUES ('Eazy Bytes','tutor@eazybytes.com','9823150966',CURDATE());

INSERT INTO accounts (customer\_id,account\_number,account\_type,branch\_address,create\_dt)

VALUES (1,654321,'Savings','123, Main Street',CURDATE());

accounts\src\main\resources\schema.sql

DROP TABLE IF EXISTS customer;

DROP TABLE IF EXISTS accounts;

CREATE TABLE customer(

customer\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) NOT NULL,

mobile\_number VARCHAR(20) NOT NULL,

create\_dt date DEFAULT NULL

);

CREATE TABLE accounts(

customer\_id INT NOT NULL,

account\_number INT AUTO\_INCREMENT PRIMARY KEY,

account\_type VARCHAR(100) NOT NULL,

branch\_address VARCHAR(200) NOT NULL,

create\_dt date DEFAULT NULL

);

Create relevant packages: controller, repository, model, exceptions

package com.eazybytes.accounts.model;

import java.time.LocalDate;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import lombok.Getter;

import lombok.Setter;

import lombok.ToString;

@Entity

@Getter @Setter @ToString

public class Accounts {

@Column(name = "customer\_id")

private int customerId;

@Column(name = "account\_number")

@Id

private long accountNumber;

@Column(name = "account\_type")

private String accountType;

@Column(name = "branch\_address")

private String branchAddress;

@Column(name = "create\_dt")

private LocalDate createDt;

}

package com.eazybytes.accounts.model;

import java.time.LocalDate;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import lombok.Getter;

import lombok.Setter;

import lombok.ToString;

@Entity

@Getter @Setter @ToString

public class Customer {

@Column(name = "customer\_id")

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private int customerId;

@Column(name = "name")

private String name;

@Column(name = "email")

private String email;

@Column(name = "mobile\_number")

private String mobileNumber;

@Column(name = "create\_dt")

private LocalDate createDt;

}

package com.eazybytes.accounts.repository;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

import com.eazybytes.accounts.model.Accounts;

@Repository

public interface AccountsRepository extends CrudRepository<Accounts, Long>{

Accounts findByCustomerId(int customerId);

}

package com.eazybytes.accounts.exceptions;

public class AccountNotFoundException extends RuntimeException {

public AccountNotFoundException() {

super();

}

public AccountNotFoundException(String message) {

super(message);

}

public AccountNotFoundException(String message, Throwable cause) {

super(message, cause);

}

public AccountNotFoundException(Throwable cause) {

super(cause);

}

}

package com.eazybytes.accounts.controller;

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

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.eazybytes.accounts.exceptions.AccountNotFoundException;

import com.eazybytes.accounts.model.Accounts;

import com.eazybytes.accounts.model.Customer;

import com.eazybytes.accounts.repository.AccountsRepository;

@RestController

public class AccountsController {

@Autowired

private AccountsRepository accountsRepository;

@PostMapping("/myAccount")

public Accounts getAccountDetails(@RequestBody Customer customer){

Accounts accounts =

accountsRepository.findByCustomerId(customer.getCustomerId());

if (accounts == null) {

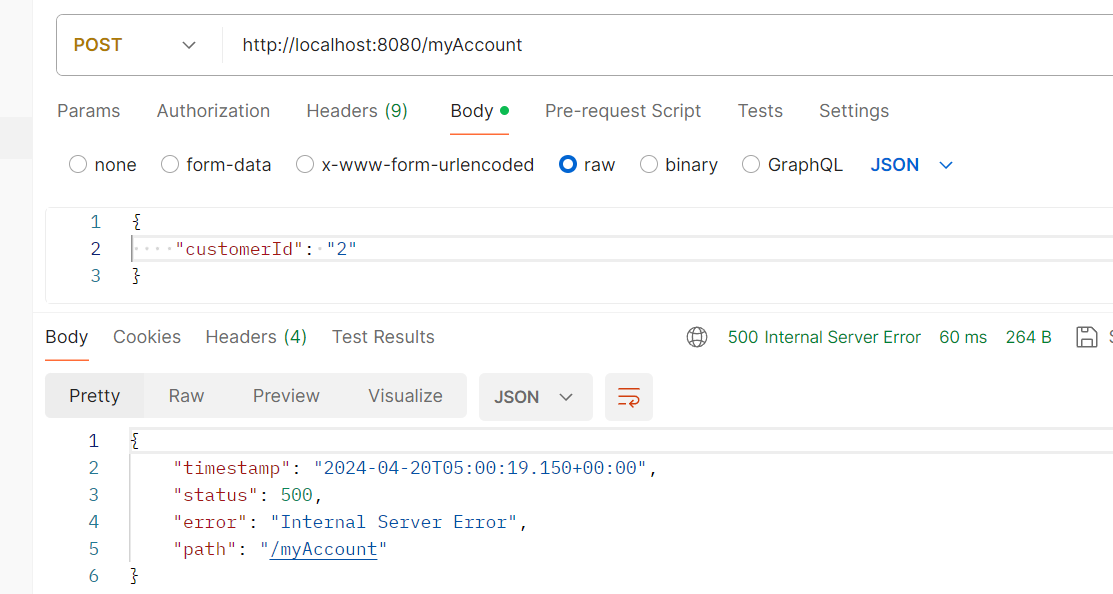
throw new AccountNotFoundException("Account not found for customer " + customer.getCustomerId());

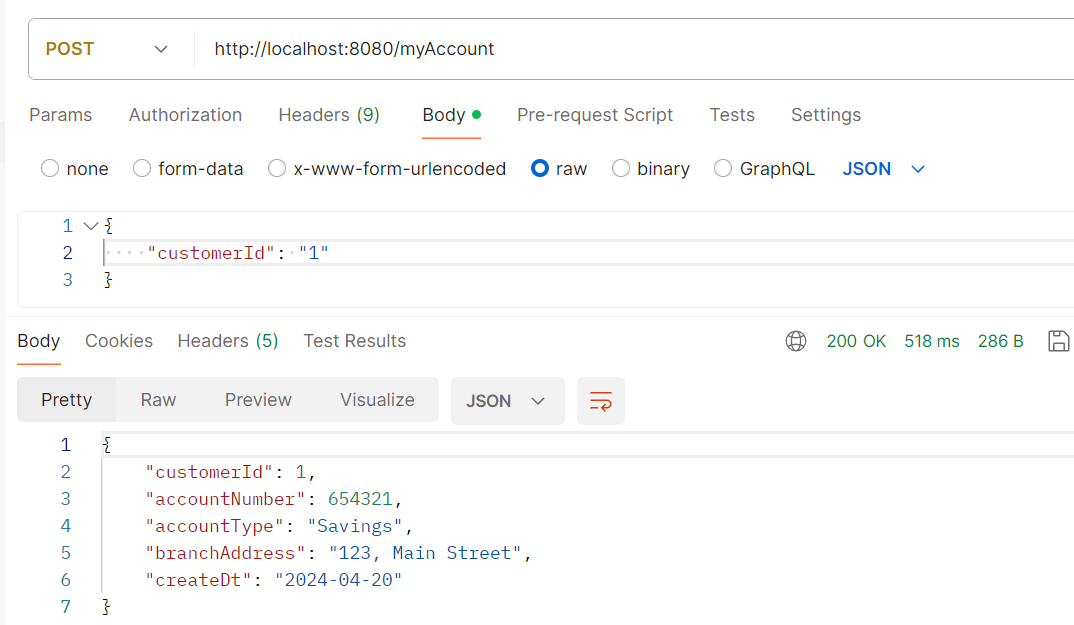
}

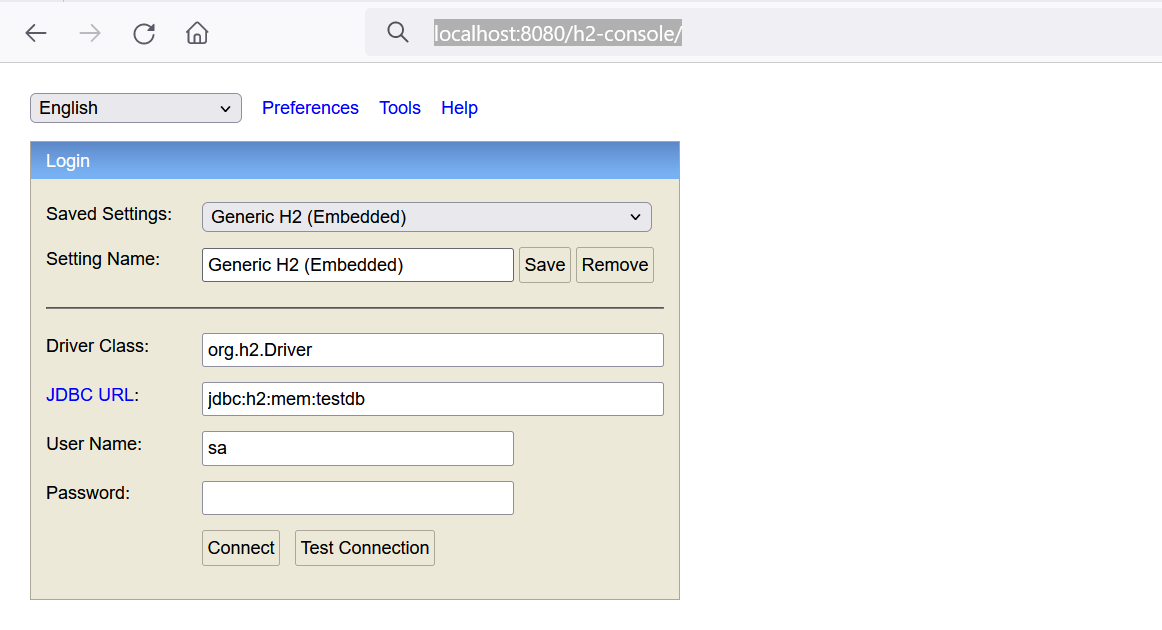
return accounts;

}

}







Step – 2

Create basic SB microservice project

<groupId>com.eazybytes</groupId>

<artifactId>loans</artifactId>

package com.eazybytes.loans;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.autoconfigure.domain.EntityScan;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.ComponentScans;

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

@SpringBootApplication

@ComponentScans({@ComponentScan("com.eazybytes.loans.controller")})

@EnableJpaRepositories("com.eazybytes.loans.repository")

@EntityScan("com.eazybytes.loans.model")

public class LoansApplication {

public static void main(String[] args) {

SpringApplication.run(LoansApplication.class, args);

}

}

loans\src\main\resources\application.properties

spring.application.name=loans

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driver-class-name=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.h2.console.enabled=true

server.port=8090

loans\src\main\resources\data.sql

INSERT INTO `loans` ( `customer\_id`, `start\_dt`, `loan\_type`, `total\_loan`, `amount\_paid`, `outstanding\_amount`, `create\_dt`)

VALUES ( 1, '2020-10-13', 'Home', 200000, 50000, 150000, '2020-10-13');

INSERT INTO `loans` ( `customer\_id`, `start\_dt`, `loan\_type`, `total\_loan`, `amount\_paid`, `outstanding\_amount`, `create\_dt`)

VALUES ( 1, '2020-06-06', 'Vehicle', 40000, 10000, 30000, '2020-06-06');

INSERT INTO `loans` ( `customer\_id`, `start\_dt`, `loan\_type`, `total\_loan`, `amount\_paid`, `outstanding\_amount`, `create\_dt`)

VALUES ( 1, '2021-02-14', 'Home', 50000, 10000, 40000, '2018-02-14');

INSERT INTO `loans` ( `customer\_id`, `start\_dt`, `loan\_type`, `total\_loan`, `amount\_paid`, `outstanding\_amount`, `create\_dt`)

VALUES ( 1, '2018-02-14', 'Personal', 10000, 3500, 6500, '2018-02-14');

loans\src\main\resources\schema.sql

DROP TABLE IF EXISTS loans;

CREATE TABLE `loans` (

`loan\_number` int NOT NULL AUTO\_INCREMENT,

`customer\_id` int NOT NULL,

`start\_dt` date NOT NULL,

`loan\_type` varchar(100) NOT NULL,

`total\_loan` int NOT NULL,

`amount\_paid` int NOT NULL,

`outstanding\_amount` int NOT NULL,

`create\_dt` date DEFAULT NULL,

PRIMARY KEY (`loan\_number`)

);

Create relevant packages: controller, repository, model, exceptions

package com.eazybytes.loans.model;

import java.util.Date;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import lombok.Getter;

import lombok.Setter;

import lombok.ToString;

@Entity

@Getter @Setter @ToString

public class Loans {

@GeneratedValue(strategy = GenerationType.AUTO)

@Column(name = "loan\_number")

@Id

private int loanNumber;

@Column(name = "customer\_id")

private int customerId;

@Column(name = "start\_dt")

private Date startDt;

@Column(name = "loan\_type")

private String loanType;

@Column(name = "total\_loan")

private int totalLoan;

@Column(name = "amount\_paid")

private int amountpaid;

@Column(name = "outstanding\_amount")

private int outstandingAmount;

@Column(name = "create\_dt")

private String createDt;

}

package com.eazybytes.loans.model;

import lombok.Getter;

import lombok.Setter;

import lombok.ToString;

@Getter @Setter @ToString

public class Customer {

private int customerId;

}

package com.eazybytes.loans.repository;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

import com.eazybytes.loans.model.Loans;

@Repository

public interface LoansRepository extends CrudRepository<Loans, Integer> {

java.util.List<Loans> findByCustomerIdOrderByStartDtDesc(int customerId);

}

package com.eazybytes.loans.exceptions;

public class LoanNotFoundException extends RuntimeException {

public LoanNotFoundException() {

super();

}

public LoanNotFoundException(String message) {

super(message);

}

public LoanNotFoundException(String message, Throwable cause) {

super(message, cause);

}

public LoanNotFoundException(Throwable cause) {

super(cause);

}

}

package com.eazybytes.loans.controller;

import java.util.List;

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

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.eazybytes.loans.exceptions.LoanNotFoundException;

import com.eazybytes.loans.model.Customer;

import com.eazybytes.loans.model.Loans;

import com.eazybytes.loans.repository.LoansRepository;

@RestController

public class LoansController {

@Autowired

private LoansRepository loansRepository;

@PostMapping("/myLoans")

public List<Loans> getLoans(@RequestBody Customer customer) {

List<Loans> loans = loansRepository.findByCustomerIdOrderByStartDtDesc(customer.getCustomerId());

if (loans == null) {

throw new LoanNotFoundException("Loans not found for customer " + customer.getCustomerId());

}

return loans;

}

}

