package com.onlinebanking.ICINBank.controllers;

import com.onlinebanking.ICINBank.dto.AccountDto;

import com.onlinebanking.ICINBank.dto.TransactionRegisterDto;

import com.onlinebanking.ICINBank.model.Account;

import com.onlinebanking.ICINBank.model.LoginRequest;

import com.onlinebanking.ICINBank.model.User;

import com.onlinebanking.ICINBank.service.ICINBankingService;

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

import org.springframework.data.repository.query.Param;

import org.springframework.ui.Model;

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

import java.util.ArrayList;

import java.util.List;

@CrossOrigin(origins="http://localhost:4200/api")

@RestController

public class ICINBankingController {

@Autowired

public ICINBankingService iCINBankingService;

@PostMapping(path="/login")

public void login(@RequestBody String username, String password, Model model){

LoginRequest loginRequest = new LoginRequest();

loginRequest.setPassword(password);

loginRequest.setUsername(username);

if(iCINBankingService.isValidUser(username, password)){

model.addAttribute(loginRequest);

}

}

@GetMapping(path="/allTransactions")

public List<TransactionRegisterDto> transactionList(@Param("toAccount")long toAccount, @Param("fromAccount")long fromAccount){

try{

return iCINBankingService.getAllTransactionsByAccountKey(toAccount, fromAccount);

}catch(Exception e){

System.out.println("Unable to fetch transactions for account " + toAccount+ "and " + " "+ fromAccount + " "+e.getMessage());

}

return new ArrayList<>();

}

@GetMapping(path="/allUsers")

public Iterable<User> userList(){

return iCINBankingService.getUserList();

}

@GetMapping(path="/allAccounts")

public Iterable<Account> accountList(){

return iCINBankingService.getAccountList();

}

@GetMapping(path="/users/{userKey}/accounts")

@ResponseBody

public List<AccountDto> getUserAccountsList(@PathVariable("userKey") String user){

try{

return iCINBankingService.getUserAccountsList(Long.parseLong(user));

}catch(Exception e){

System.out.println("Unable to fetch accounts for user " + user+ " "+ e.getMessage());

}

return new ArrayList<>();

}

}

package com.onlinebanking.ICINBank.config;

import org.springframework.boot.context.properties.ConfigurationProperties;

import org.springframework.boot.jdbc.DataSourceBuilder;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import javax.sql.DataSource;

import java.sql.SQLException;

@Configuration

@ConfigurationProperties(prefix="spring.datasource")

public class JpaConfig {

@Bean

public DataSource getDataSource() throws SQLException {

return

DataSourceBuilder.create()

.driverClassName("com.mysql.cj.jdbc.Driver")

.url("jdbc:mysql://localhost:3306/onlinebanking")

.username("onlinebankinguser")

.password("password")

.build();

}

}

package com.onlinebanking.ICINBank.dto;

import com.onlinebanking.ICINBank.enums.AccountType;

import lombok.Getter;

import lombok.Setter;

@Setter

@Getter

public class AccountDto {

private String accountType;

private int balance;

}

package com.onlinebanking.ICINBank.dto;

import lombok.Getter;

import lombok.Setter;

import java.time.LocalDate;

@Setter

@Getter

public class TransactionRegisterDto {

private double amount;

private String transactionType;

private LocalDate timestamp;

private long fromAccount;

private long toAccount;

}

package com.onlinebanking.ICINBank.enums;

import lombok.Getter;

@Getter

public enum AccountType {

CHECKING("checking"),

SAVING("saving");

private final String type;

AccountType(String type) {

this.type=type;

}

}

package com.onlinebanking.ICINBank.enums;

import lombok.Getter;

@Getter

public enum TransactionType {

WITHDRAW("withdraw"),

TRANSFER("transfer"),

DEPOSIT("deposit"),

CHECK\_ORDER("checkOrder");

private final String type;

TransactionType(String type) {

this.type=type;

}

}

package com.onlinebanking.ICINBank.model;

import com.onlinebanking.ICINBank.enums.AccountType;

import lombok.Getter;

import lombok.Setter;

import org.hibernate.validator.internal.util.stereotypes.Lazy;

import javax.persistence.\*;

import java.util.List;

@Entity

@Getter

@Setter

public class Account {

@Id

@GeneratedValue(strategy= GenerationType.AUTO)

@Column(name="ACCOUNT\_KEY")

private long accountKey;

// @ManyToOne

// private User user;

@Column(name="BALANCE")

private int balance;

@OneToMany

@JoinColumn(name="fromAccount")

private List<TransactionRegister> listOfFromTransactions;

@OneToMany

@JoinColumn(name="toAccount")

private List<TransactionRegister> listOfToTransactions;

@Column(name="ACCOUNT\_TYPE")

private AccountType accountType;

public Account() {

}

public Account(User accountHolderUser, int balance, AccountType accountType) {

// this.user = accountHolderUser;

this.balance = balance;

this.accountType = accountType;

}

@Override

public String toString() {

return "Account{" +

"accountKey=" + accountKey +

// ", accountHolderUser=" + accountHolderUser +

", balance=" + balance +

", accountType=" + accountType +

'}';

}

}

package com.onlinebanking.ICINBank.model;

import lombok.Getter;

import lombok.Setter;

@Setter

@Getter

public class LoginRequest {

private String username;

private String password;

}

package com.onlinebanking.ICINBank.model;

import com.onlinebanking.ICINBank.enums.TransactionType;

import lombok.Getter;

import lombok.Setter;

import javax.persistence.\*;

import java.time.LocalDate;

@Entity

@Getter

@Setter

public class TransactionRegister {

@Id

@GeneratedValue(strategy= GenerationType.AUTO)

@Column(name="TRANSACTION\_REGISTER\_KEY")

private long transactionRegisterKey;

//

// @ManyToOne

//// @JoinColumn(name="FROM\_ACCOUNT", nullable=false, updatable=false)

// private Account fromAccount;

// @ManyToOne

//// @JoinColumn(name="TO\_ACCOUNT", nullable=false, updatable=false)

// private Account toAccount;

@Column(name="TIMESTAMP")

private LocalDate timestamp;

@Column(name="TRANSACTION\_TYPE")

private TransactionType transactionType;

@Column(name="AMOUNT")

private double amount;

public TransactionRegister( LocalDate timestamp, TransactionType transactionType, double amount) {

// this.fromAccount = fromAccount;

// this.toAccount = toAccount;

this.timestamp = timestamp;

this.transactionType = transactionType;

this.amount = amount;

}

public TransactionRegister(){

}

@Override

public String toString() {

return "TransactionRegister{" +

"transactionRegisterKey=" + transactionRegisterKey +

// ", fromAccount=" + fromAccount +

// ", toAccount=" + toAccount +

", timestamp=" + timestamp +

", transactionType=" + transactionType +

'}';

}

}

package com.onlinebanking.ICINBank.model;

import lombok.Getter;

import lombok.Setter;

import javax.persistence.\*;

import java.util.List;

@Entity

@Getter

@Setter

public class User {

@Id

@GeneratedValue(strategy= GenerationType.AUTO)

@Column(name="USER\_KEY")

private long userKey;

@Column(name="FIRST\_NAME")

private String firstName;

@Column(name="LAST\_NAME")

private String lastName;

@OneToMany(cascade=CascadeType.ALL)

@JoinColumn(name="userKey")

private List<Account> accounts;

@Column(name="EMAIL")

private String email;

@Column(name="PHONE")

private String phone;

@Column(name="USERNAME")

private String username;

@Column(name="password")

private String password;

@Column(name="PIN")

private int pin;

public User() {

}

public User(String firstName, String lastName, List<Account> accounts, String email, String phone, String username, int pin) {

this.firstName = firstName;

this.lastName = lastName;

this.accounts = accounts;

this.email = email;

this.phone = phone;

this.username = username;

this.pin = pin;

}

@Override

public String toString() {

return "User{" +

"userKey=" + userKey +

", firstName='" + firstName + '\'' +

", lastName='" + lastName + '\'' +

", accounts=" + accounts +

", email='" + email + '\'' +

", phone='" + phone + '\'' +

", username='" + username + '\'' +

", pin=" + pin +

'}';

}

}

package com.onlinebanking.ICINBank.repository;

import com.onlinebanking.ICINBank.model.Account;

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

import org.springframework.data.repository.CrudRepository;

import org.springframework.data.repository.query.Param;

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import java.util.List;

@Repository

public interface AccountRepository extends CrudRepository<Account, Long> {

@Transactional(readOnly=true)

@Query(value= "SELECT ac.account\_type, ac.balance FROM account ac WHERE"

+ " ac.user\_key =:userKey",nativeQuery = true)

List<Long> findAllAccountsByUserKey(@Param("userKey")long userKey);

}

package com.onlinebanking.ICINBank.repository;

import com.onlinebanking.ICINBank.model.TransactionRegister;

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

import org.springframework.data.repository.CrudRepository;

import org.springframework.data.repository.query.Param;

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import java.util.List;

@Repository

public interface TransactionRegisterRepository extends CrudRepository<TransactionRegister, Long> {

@Transactional(readOnly=true)

@Query(value= "SELECT tr.from\_account, tr.to\_account, tr.amount, tr.transaction\_type, tr.timestamp FROM transaction\_register tr WHERE"

+ " tr.to\_account =:toAccount or tr.to\_account =:fromAccount",nativeQuery = true)

List<Long> findListOfTransactionsByAccountKey(@Param("toAccount")long toAccount, @Param("fromAccount")long fromAccount);

}

package com.onlinebanking.ICINBank.repository;

import com.onlinebanking.ICINBank.model.User;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface UserRepository extends CrudRepository<User, Long> {

public User findByUsername(String username);

public User getUserByUserKey(long userKey);

}

package com.onlinebanking.ICINBank.service;

import com.onlinebanking.ICINBank.dto.AccountDto;

import com.onlinebanking.ICINBank.dto.TransactionRegisterDto;

import com.onlinebanking.ICINBank.model.Account;

import com.onlinebanking.ICINBank.model.TransactionRegister;

import com.onlinebanking.ICINBank.model.User;

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

import java.time.LocalDate;

import java.util.List;

public interface ICINBankingService {

public Iterable<TransactionRegister> transactionList(@RequestBody long accountId, LocalDate date);

public List<TransactionRegisterDto> getAllTransactionsByAccountKey(long toAccount, long fromAccount);

public List<AccountDto> findAllAccountsByUserKey(long userKey);

public Iterable<User> getUserList();

public Iterable<Account> getAccountList();

public boolean isValidUser(String username, String password);

List<AccountDto> getUserAccountsList(long userKey);

}

package com.onlinebanking.ICINBank.service;

import com.onlinebanking.ICINBank.dto.AccountDto;

import com.onlinebanking.ICINBank.dto.TransactionRegisterDto;

import com.onlinebanking.ICINBank.enums.AccountType;

import com.onlinebanking.ICINBank.model.Account;

import com.onlinebanking.ICINBank.model.TransactionRegister;

import com.onlinebanking.ICINBank.model.User;

import com.onlinebanking.ICINBank.repository.AccountRepository;

import com.onlinebanking.ICINBank.repository.TransactionRegisterRepository;

import com.onlinebanking.ICINBank.repository.UserRepository;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import java.time.LocalDate;

import java.util.ArrayList;

import java.util.List;

import java.util.Objects;

import java.util.Optional;

import java.util.stream.Collectors;

import java.util.stream.IntStream;

@Service

public class ICINBankingServiceImpl implements ICINBankingService{

@Autowired

private AccountRepository accountRepository;

@Autowired

private UserRepository userRepository;

@Autowired

private TransactionRegisterRepository transactionRegisterRepository;

@Override

public boolean isValidUser(String username, String password){

if(!username.isEmpty() && !password.isEmpty()){

if(userRepository.findByUsername(username) != null){

if(Objects.equals(userRepository.findByUsername(username).getPassword(), password)){

return true;

}

}

}

return false;

}

@Override

public List<AccountDto> getUserAccountsList(long userKey) {

return findAllAccountsByUserKey(userKey);

}

private User getUserByUserKey(long userKey){

return userRepository.getUserByUserKey(userKey);

}

@Override

public Iterable<User> getUserList(){

return userRepository.findAll();

}

@Override

public Iterable<Account> getAccountList() {

return accountRepository.findAll();

}

@Override

public List<TransactionRegisterDto> getAllTransactionsByAccountKey(long toAccount, long fromAccount){

List<Long> listOfTransactions = transactionRegisterRepository.findListOfTransactionsByAccountKey(toAccount,fromAccount);

List<TransactionRegisterDto> transactionRegisterDtoList = new ArrayList<TransactionRegisterDto>();

IntStream.range(1, listOfTransactions.size() + 1).mapToObj(i -> this.setTransactionRegisterDtoByTransactionId((long) i)).collect(Collectors.toList());

return transactionRegisterDtoList;

}

@Transactional

private TransactionRegisterDto setTransactionRegisterDtoByTransactionId(Long accountId) { //add the secong from account check

Optional<TransactionRegister> transactionRegister = transactionRegisterRepository.findById(accountId);

TransactionRegisterDto transactionRegisterDto = new TransactionRegisterDto();

transactionRegisterDto.setAmount(transactionRegister.get().getAmount());

transactionRegisterDto.setTimestamp(transactionRegister.get().getTimestamp());

transactionRegisterDto.setTransactionType(transactionRegister.get().getTransactionType().getType());

transactionRegisterDto.setToAccount(accountRepository.findById(accountId).get().getAccountKey());

transactionRegisterDto.setFromAccount(accountRepository.findById(accountId).get().getAccountKey());

return transactionRegisterDto;

}

@Override

public List<AccountDto> findAllAccountsByUserKey(long userKey) {

List<Long> listOfAccounts = accountRepository.findAllAccountsByUserKey(userKey);

List<AccountDto> accountDtoList = IntStream.range(1, listOfAccounts.size() + 1).mapToObj(i -> this.getDtoByAccountId((long) i)).collect(Collectors.toList());

return accountDtoList;

}

@Transactional

private AccountDto getDtoByAccountId(Long accountId) {

Optional<Account> account = accountRepository.findById(accountId);

AccountDto accountDto = new AccountDto();

accountDto.setAccountType(account.get().getAccountType().getType());

accountDto.setBalance(account.get().getBalance());

return accountDto;

}

@Override

public Iterable<TransactionRegister> transactionList(long accountId, LocalDate date) {

return null;

}

}

package com.onlinebanking.ICINBank;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import springfox.documentation.swagger2.annotations.EnableSwagger2WebMvc;

@SpringBootApplication

public class IcinBankApplication {

public static void main(String[] args) {

SpringApplication.run(IcinBankApplication.class, args);

}

}

package com.onlinebanking.ICINBank.tests;

import org.junit.jupiter.api.Test;

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

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.boot.test.web.client.TestRestTemplate;

import org.springframework.boot.web.server.LocalServerPort;

import static org.assertj.core.api.Assertions.assertThat;

@SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.RANDOM\_PORT)

public class LoginTest {

@LocalServerPort

private int port;

@Autowired

private TestRestTemplate restTemplate;

@Test

public void checkIfUserCanLogin() throws Exception {

assertThat(this.restTemplate.getForObject("http://localhost:" + port + "/",

String.class)).contains("Login");

}

}

package com.onlinebanking.ICINBank;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

class IcinBankApplicationTests {

@Test

void contextLoads() {

}

}