**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе №3

по дисциплине «**Технологии программирования**»

Автор: Юрпалов Сергей Николаевич

Факультет: ИТиП

Группа: М32051



Санкт-Петербург 2022

Описание:

К созданному в прошлой лабораторной сервису добавляется Spring.

Сервис должен предоставлять http интерфейс (REST API) для получения информации о конкретных котиках и владельцах и для получения фильтрованной информации (например, получить всех рыжих котиков)

Внимание: недопустимо отдавать через HTTP интерфейс сущности JPA. Рекомендуется создать отдельные оберточные классы.

Сервисы и dao должны превратиться в Spring Bean’ы с использованием Dependency Injection (Autowired). Dao при этом наследуют JpaRepository и имеет шаблонные Spring Data JPA методы: https://www.baeldung.com/spring-data-repositories#repositories

При сдаче лабораторной нужно будет показать работоспособность endpoint’ов через http запросы (рекомендуется Postman).

В рамках лабораторной к проекту должен быть добавлен CI/CD, запускающий тесты проекта kotiki-java.

Исходный код:

Изменен файл kotiki-java/database-access/pom.xml:

-<?xml version="1.0" encoding="UTF-8"?>

+<?xml version="1.0"?>

<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">

@@ -25,7 +25,7 @@

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

- <version>5.6.5.Final</version>

+ <version>5.6.7.Final</version>

</dependency>

</dependencies>

Изменен файл kotiki-java/database-access/src/main/java/dao/CatDao.java:

package dao;

-import dao.interfaces.AbstractDao;

import models.Cat;

-import org.hibernate.Session;

-import org.hibernate.Transaction;

-import tools.CatsException;

-import utils.HibernateSessionFactoryUtil;

-import java.util.List;

-

-public class CatDao implements AbstractDao<Cat> {

-

- public Cat findById(int id) {

- return HibernateSessionFactoryUtil.getSessionFactory().openSession().get(Cat.class, id);

- }

-

- public void save(Cat cat) {

- Session session = HibernateSessionFactoryUtil.getSessionFactory().openSession();

- Transaction transaction = session.beginTransaction();

- session.save(cat);

- transaction.commit();

- session.close();

- }

-

- public void update(Cat cat) {

- Session session = HibernateSessionFactoryUtil.getSessionFactory().openSession();

- Transaction transaction = session.beginTransaction();

- session.merge(cat);

- transaction.commit();

- session.close();

- }

-

- public void delete(int id) throws CatsException {

- Session session = HibernateSessionFactoryUtil.getSessionFactory().openSession();

- Transaction transaction = session.beginTransaction();

- Cat cat = session.get(Cat.class, id);

- for (Cat friend : cat.getFriends()) {

- friend.removeFriend(cat);

- }

- session.delete(cat);

- transaction.commit();

- session.close();

- }

+import models.enums.CatColor;

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

+import java.time.LocalDate;

+import java.util.List;

- public List<Cat> findAll() {

- return HibernateSessionFactoryUtil

- .getSessionFactory()

- .openSession()

- .createQuery("FROM Cat", Cat.class)

- .list();

- }

+public interface CatDao extends JpaRepository<Cat, Integer> {

+ List<Cat> findByName(String name);

+ List<Cat> findByBirthDate(LocalDate birthDate);

+ List<Cat> findByBreed(String breed);

+ List<Cat> findByColor(CatColor color);

}

Изменен файл kotiki-java/database-access/src/main/java/dao/OwnerDao.java:

package dao;

-import dao.interfaces.AbstractDao;

import models.Owner;

-import org.hibernate.Session;

-import org.hibernate.Transaction;

-import utils.HibernateSessionFactoryUtil;

-import java.util.List;

-

-public class OwnerDao implements AbstractDao<Owner> {

-

- public Owner findById(int id) {

- return HibernateSessionFactoryUtil.openSession().get(Owner.class, id);

- }

-

- public void save(Owner owner) {

- Session session = HibernateSessionFactoryUtil.openSession();

- Transaction transaction = session.beginTransaction();

- session.save(owner);

- transaction.commit();

- session.close();

- }

-

- public void update(Owner owner) {

- Session session = HibernateSessionFactoryUtil.openSession();

- Transaction transaction = session.beginTransaction();

- session.merge(owner);

- transaction.commit();

- session.close();

- }

-

- public void delete(int id) {

- Session session = HibernateSessionFactoryUtil.openSession();

- Transaction transaction = session.beginTransaction();

- Owner owner = session.get(Owner.class, id);

- session.delete(owner);

- transaction.commit();

- session.close();

- }

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

+import java.time.LocalDate;

+import java.util.List;

- public List<Owner> findAll() {

- return HibernateSessionFactoryUtil

- .getSessionFactory()

- .openSession()

- .createQuery("FROM Owner", Owner.class)

- .list();

- }

+public interface OwnerDao extends JpaRepository<Owner, Integer> {

+ List<Owner> findByName(String name);

+ List<Owner> findByBirthDate(LocalDate birthDate);

}

Изменен файл kotiki-java/database-access/src/main/java/models/Cat.java:

package models;

-import tools.CatsException;

+import models.enums.CatColor;

import javax.persistence.\*;

import java.time.LocalDate;

public class Cat {

@Enumerated(EnumType.STRING)

@Column(name = "color")

- private CatColors color;

+ private CatColor color;

@ManyToOne(fetch = FetchType.LAZY)

@JoinTable(

public class Cat {

public Cat() {}

- public Cat(String name, LocalDate birthDate, String breed, CatColors color) {

+ public Cat(String name, LocalDate birthDate, String breed, CatColor color) {

this.name = name;

this.birthDate = birthDate;

this.breed = breed;

public void setBreed(String breed) {

this.breed = breed;

}

- public CatColors getColor() {

+ public CatColor getColor() {

return color;

}

- public void setColor(CatColors color) {

+ public void setColor(CatColor color) {

this.color = color;

}

public void addFriend(Cat cat) throws CatsException {

if (equals(cat)) {

throw new CatsException("Cat cannot make friends with himself");

}

+

friendOf.add(cat);

friends.add(cat);

+ cat.getFriends().add(this);

+ cat.getFriendOf().add(this);

}

public void removeFriend(Cat cat) throws CatsException {

if (!getFriends().contains(cat)) {

throw new CatsException("Cat does not have such friend");

}

+

friendOf.remove(cat);

friends.remove(cat);

+ cat.getFriends().remove(this);

+ cat.getFriendOf().remove(this);

}

@Override

Добавлен файл kotiki-java/database-access/src/main/java/models/enums/CatColor.java:

package models.enums;

public enum CatColor {

WHITE,

BLACK,

RED,

GREY,

BROWN

}

Изменен файл kotiki-java/pom.xml:

-<?xml version="1.0" encoding="UTF-8"?>

+<?xml version="1.0"?>

<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">

@@ -10,24 +10,41 @@

<version>1.0-SNAPSHOT</version>

<modules>

<module>database-access</module>

+ <module>user-service</module>

</modules>

<dependencies>

- <dependency>

- <groupId>org.wilfordaf</groupId>

- <artifactId>database-access</artifactId>

- <version>1.0-SNAPSHOT</version>

- <scope>compile</scope>

- </dependency>

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter-api</artifactId>

- <version>5.6.2</version>

+ <version>5.8.2</version>

+ <scope>test</scope>

+ </dependency>

+ <dependency>

+ <groupId>org.mockito</groupId>

+ <artifactId>mockito-core</artifactId>

+ <version>4.4.0</version>

<scope>test</scope>

</dependency>

<dependency>

- <groupId>org.junit.vintage</groupId>

- <artifactId>junit-vintage-engine</artifactId>

- <version>5.6.2</version>

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

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

+ <version>2.6.5</version>

+ </dependency>

+ <dependency>

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

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

+ <version>2.6.5</version>

+ </dependency>

+ <dependency>

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

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

+ <version>2.6.5</version>

+ <scope>runtime</scope>

+ </dependency>

+ <dependency>

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

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

+ <version>2.6.5</version>

<scope>test</scope>

</dependency>

</dependencies>

@@ -37,4 +54,13 @@

<maven.compiler.target>17</maven.compiler.target>

</properties>

+ <build>

+ <plugins>

+ <plugin>

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

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

+ <version>2.6.5</version>

+ </plugin>

+ </plugins>

+ </build>

</project>

Добавлен файл kotiki-java/user-service/pom.xml:

<?xml version="1.0"?>

<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">

<parent>

<artifactId>java-kotiki</artifactId>

<groupId>org.wilfordaf</groupId>

<version>1.0-SNAPSHOT</version>

</parent>

<modelVersion>4.0.0</modelVersion>

<artifactId>user-service</artifactId>

<dependencies>

<dependency>

<groupId>org.wilfordaf</groupId>

<artifactId>database-access</artifactId>

<version>1.0-SNAPSHOT</version>

<scope>compile</scope>

</dependency>

</dependencies>

<properties>

<maven.compiler.source>17</maven.compiler.source>

<maven.compiler.target>17</maven.compiler.target>

</properties>

</project>

Добавлен файл kotiki-java/user-service/src/main/java/controllers/CatController.java:

package controllers;

import dto.CatDto;

import dto.mappers.CatToDtoMapper;

import dto.mappers.DtoToCatMapper;

import models.Cat;

import models.enums.CatColor;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

import org.springframework.web.server.ResponseStatusException;

import services.CatService;

import java.time.LocalDate;

import java.util.List;

@RestController

@RequestMapping("/cats")

public class CatController {

private final CatService catService;

private final CatToDtoMapper catToDtoMapper;

private final DtoToCatMapper dtoToCatMapper;

@Autowired

public CatController(CatService catService, CatToDtoMapper catToDtoMapper, DtoToCatMapper dtoToCatMapper) {

this.catService = catService;

this.catToDtoMapper = catToDtoMapper;

this.dtoToCatMapper = dtoToCatMapper;

}

@GetMapping("/get/{id}")

public ResponseEntity<CatDto> getById(@PathVariable int id) {

Cat cat = catService.getById(id);

if (cat == null) {

return new ResponseEntity<>(HttpStatus.NOT\_FOUND);

}

return ResponseEntity.ok(catToDtoMapper.convert(cat));

}

@PostMapping("/save")

public void save(@RequestBody CatDto catDto) {

catService.save(dtoToCatMapper.convert(catDto));

}

@PostMapping("/delete/{id}")

public void deleteById(@PathVariable int id) {

try {

catService.deleteById(id);

}

catch (Exception \_\_) {

throw new ResponseStatusException(HttpStatus.BAD\_REQUEST);

}

}

@GetMapping ("/find-all")

public List<CatDto> findAll() {

return catService.findAll().stream().map(catToDtoMapper::convert).toList();

}

@GetMapping("/find-by-name/{name}")

public List<CatDto> findByName(@PathVariable String name) {

return catService.findByName(name).stream().map(catToDtoMapper::convert).toList();

}

@GetMapping("/find-by-birthDate/{birthDate}")

public List<CatDto> findByBirthDate(@PathVariable String birthDate) {

LocalDate date = LocalDate.parse(birthDate);

return catService.findByBirthDate(date).stream().map(catToDtoMapper::convert).toList();

}

@GetMapping("/find-by-breed/{breed}")

public List<CatDto> findByBreed(@PathVariable String breed) {

return catService.findByBreed(breed).stream().map(catToDtoMapper::convert).toList();

}

@GetMapping("/find-by-color/{color}")

public List<CatDto> findByColor(@PathVariable CatColor color) {

return catService.findByColor(color).stream().map(catToDtoMapper::convert).toList();

}

}

Добавлен файл kotiki-java/user-service/src/main/java/controllers/OwnerController.java:

package controllers;

import dto.OwnerDto;

import dto.mappers.DtoToOwnerMapper;

import dto.mappers.OwnerToDtoMapper;

import models.Owner;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

import org.springframework.web.server.ResponseStatusException;

import services.OwnerService;

import java.time.LocalDate;

import java.util.List;

@RestController

@RequestMapping("/owners")

public class OwnerController {

private final OwnerService ownerService;

private final OwnerToDtoMapper ownerToDtoMapper;

private final DtoToOwnerMapper dtoToOwnerMapper;

@Autowired

public OwnerController(OwnerService ownerService, OwnerToDtoMapper ownerToDtoMapper, DtoToOwnerMapper dtoToOwnerMapper) {

this.ownerService = ownerService;

this.ownerToDtoMapper = ownerToDtoMapper;

this.dtoToOwnerMapper = dtoToOwnerMapper;

}

@GetMapping("/get/{id}")

public ResponseEntity<OwnerDto> getById(@PathVariable int id) {

Owner owner = ownerService.getById(id);

if (owner == null) {

return new ResponseEntity<>(HttpStatus.NOT\_FOUND);

}

return ResponseEntity.ok(ownerToDtoMapper.convert(owner));

}

@PostMapping("/update")

public void save(@RequestBody OwnerDto ownerDto) {

ownerService.save(dtoToOwnerMapper.convert(ownerDto));

}

@PostMapping("/delete/{id}")

public void deleteById(@PathVariable int id) {

try {

ownerService.deleteById(id);

}

catch (Exception \_\_) {

throw new ResponseStatusException(HttpStatus.BAD\_REQUEST);

}

}

@GetMapping ("/find-all")

public List<OwnerDto> findAll() {

return ownerService.findAll().stream().map(ownerToDtoMapper::convert).toList();

}

@GetMapping ("/find-by-name/{name}")

public List<OwnerDto> findByName(@PathVariable String name) {

return ownerService.findByName(name).stream().map(ownerToDtoMapper::convert).toList();

}

@GetMapping ("/find-by-birthDate/{birthDate}")

public List<OwnerDto> findByBirthDate(@PathVariable String birthDate) {

LocalDate date = LocalDate.parse(birthDate);

return ownerService.findByBirthDate(date).stream().map(ownerToDtoMapper::convert).toList();

}

}

Добавлен файл kotiki-java/user-service/src/main/java/dto/CatDto.java:

package dto;

import models.Owner;

import models.enums.CatColor;

import java.time.LocalDate;

import java.util.\*;

public class CatDto {

private int id;

private String name;

private LocalDate birthDate;

private String breed;

private CatColor color;

private OwnerDto ownerDto;

private Set<Integer> friends;

public CatDto() {

friends = new HashSet<>();

}

public CatDto(int id, String name, LocalDate birthDate, String breed, CatColor color, OwnerDto ownerDto, Set<Integer> friends) {

this.id = id;

this.name = name;

this.birthDate = birthDate;

this.breed = breed;

this.color = color;

this.ownerDto = ownerDto;

this.friends = friends;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public LocalDate getBirthDate() {

return birthDate;

}

public void setBirthDate(LocalDate birthDate) {

this.birthDate = birthDate;

}

public String getBreed() {

return breed;

}

public void setBreed(String breed) {

this.breed = breed;

}

public CatColor getColor() {

return color;

}

public void setColor(CatColor color) {

this.color = color;

}

public OwnerDto getOwner() {

return ownerDto;

}

public void setOwner(OwnerDto ownerDto) {

this.ownerDto = ownerDto;

}

public Set<Integer> getFriends() {

return friends;

}

public void setFriends(Set<Integer> friends) {

this.friends = friends;

}

}

Добавлен файл kotiki-java/user-service/src/main/java/dto/OwnerDto.java:

package dto;

import java.time.LocalDate;

import java.util.HashSet;

import java.util.Set;

public class OwnerDto {

private int id;

private String name;

private LocalDate birthDate;

private Set<Integer> cats;

public OwnerDto() {

cats = new HashSet<>();

}

public OwnerDto(int id, String name, LocalDate birthDate, Set<Integer> cats) {

this.id = id;

this.name = name;

this.birthDate = birthDate;

this.cats = cats;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public LocalDate getBirthDate() {

return birthDate;

}

public void setBirthDate(LocalDate birthDate) {

this.birthDate = birthDate;

}

public Set<Integer> getCats() {

return cats;

}

public void setCats(Set<Integer> cats) {

this.cats = cats;

}

}

Добавлен файл kotiki-java/user-service/src/main/java/dto/mappers/CatToDtoMapper.java:

package dto.mappers;

import dto.CatDto;

import models.Cat;

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

import org.springframework.stereotype.Component;

import java.util.stream.Collectors;

@Component

public class CatToDtoMapper {

private final OwnerToDtoMapper ownerToDtoMapper;

@Autowired

public CatToDtoMapper(OwnerToDtoMapper ownerToDtoMapper) {

this.ownerToDtoMapper = ownerToDtoMapper;

}

public CatDto convert(Cat cat) {

CatDto catDto = new CatDto();

catDto.setId(cat.getId());

catDto.setName(cat.getName());

catDto.setBirthDate(cat.getBirthDate());

catDto.setBreed(cat.getBreed());

catDto.setColor(cat.getColor());

catDto.setOwner(null);

if (cat.getOwner() != null) {

catDto.setOwner(ownerToDtoMapper.convert(cat.getOwner()));

}

catDto.setFriends(cat.getFriends().stream().map(Cat::getId).collect(Collectors.toSet()));

return catDto;

}

}

Добавлен файл kotiki-java/user-service/src/main/java/dto/mappers/DtoToCatMapper.java:

package dto.mappers;

import dto.CatDto;

import models.Cat;

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

import org.springframework.stereotype.Component;

import services.CatService;

import java.util.HashSet;

import java.util.Set;

import java.util.stream.Collectors;

@Component

public class DtoToCatMapper {

private final CatService catService;

private final DtoToOwnerMapper dtoToOwnerMapper;

@Autowired

public DtoToCatMapper(CatService catService, DtoToOwnerMapper dtoToOwnerMapper) {

this.catService = catService;

this.dtoToOwnerMapper = dtoToOwnerMapper;

}

public Cat convert(CatDto catDto) {

Cat cat = new Cat();

cat.setId(catDto.getId());

cat.setName(catDto.getName());

cat.setBirthDate(catDto.getBirthDate());

cat.setBreed(catDto.getBreed());

cat.setColor(catDto.getColor());

cat.setOwner(dtoToOwnerMapper.convert(catDto.getOwner()));

Set<Cat> friends = new HashSet<>();

if (catDto.getFriends() != null) {

friends = catDto.getFriends().stream().map(catService::getById).collect(Collectors.toSet());

}

cat.setFriends(friends);

cat.setFriendOf(friends);

return cat;

}

}

Добавлен файл kotiki-java/user-service/src/main/java/dto/mappers/DtoToOwnerMapper.java:

package dto.mappers;

import dto.CatDto;

import dto.OwnerDto;

import models.Cat;

import models.Owner;

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

import org.springframework.stereotype.Component;

import services.CatService;

import java.util.HashSet;

import java.util.Set;

import java.util.stream.Collectors;

@Component

public class DtoToOwnerMapper {

private final CatService catService;

@Autowired

public DtoToOwnerMapper(CatService catService) {

this.catService = catService;

}

public Owner convert(OwnerDto ownerDto) {

Owner owner = new Owner();

owner.setId(ownerDto.getId());

owner.setName(owner.getName());

owner.setBirthDate(owner.getBirthDate());

Set<Cat> cats = new HashSet<>();

if (ownerDto.getCats() != null) {

cats = ownerDto.getCats().stream().map(catService::getById).collect(Collectors.toSet());

}

owner.setCats(cats);

return owner;

}

}

Добавлен файл kotiki-java/user-service/src/main/java/dto/mappers/OwnerToDtoMapper.java:

package dto.mappers;

import dto.OwnerDto;

import models.Cat;

import models.Owner;

import org.springframework.stereotype.Component;

import java.util.stream.Collectors;

@Component

public class OwnerToDtoMapper {

public OwnerDto convert(Owner owner) {

OwnerDto ownerDto = new OwnerDto();

ownerDto.setId(owner.getId());

ownerDto.setName(owner.getName());

ownerDto.setBirthDate(owner.getBirthDate());

ownerDto.setCats(owner.getCats().stream().map(Cat::getId).collect(Collectors.toSet()));

return ownerDto;

}

}

Добавлен файл kotiki-java/user-service/src/main/java/services/CatService.java:

package services;

import dao.CatDao;

import models.Cat;

import models.Owner;

import models.enums.CatColor;

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

import org.springframework.stereotype.Service;

import java.time.LocalDate;

import java.util.List;

@Service

public class CatService {

private final CatDao catDao;

@Autowired

public CatService(CatDao catDao) {

this.catDao = catDao;

}

public Cat getById(int id) {

return catDao.getById(id);

}

public void save(Cat cat) {

catDao.save(cat);

}

public void deleteById(int id) {

catDao.deleteById(id);

}

public List<Cat> findAll() {

return catDao.findAll();

}

public List<Cat> findByName(String name) {

return catDao.findByName(name);

}

public List<Cat> findByBirthDate(LocalDate birthDate) {

return catDao.findByBirthDate(birthDate);

}

public List<Cat> findByBreed(String breed) {

return catDao.findByBreed(breed);

}

public List<Cat> findByColor(CatColor color) {

return catDao.findByColor(color);

}

}

Добавлен файл kotiki-java/user-service/src/main/java/services/OwnerService.java:

package services;

import dao.OwnerDao;

import models.Owner;

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

import org.springframework.stereotype.Service;

import java.time.LocalDate;

import java.util.List;

@Service

public class OwnerService {

private final OwnerDao ownerDao;

@Autowired

public OwnerService(OwnerDao ownerDao) {

this.ownerDao = ownerDao;

}

public Owner getById(int id) {

return ownerDao.getById(id);

}

public void save(Owner owner) {

ownerDao.save(owner);

}

public void deleteById(int id) {

ownerDao.deleteById(id);

}

public List<Owner> findAll() {

return ownerDao.findAll();

}

public List<Owner> findByName(String name) {

return ownerDao.findByName(name);

}

public List<Owner> findByBirthDate(LocalDate birthDate) {

return ownerDao.findByBirthDate(birthDate);

}

}

Добавлен файл kotiki-java/user-service/src/main/java/startapp/CatsSpringApplication.java:

package startapp;

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.data.jpa.repository.config.EnableJpaRepositories;

@SpringBootApplication

@EnableJpaRepositories("dao")

@ComponentScan({"controllers", "services", "dto"})

@EntityScan("models")

public class CatsSpringApplication {

public static void main(String[] args) {

SpringApplication.run(CatsSpringApplication.class, args);

}

}

Добавлен файл kotiki-java/user-service/src/main/resources/application.properties:

spring.datasource.url=jdbc:postgresql://localhost:5432/postgres

spring.datasource.username=postgres

spring.datasource.password=123456

Добавлен файл kotiki-java/user-service/src/test/java/services/CatServiceTest.java:

package services;

import dao.CatDao;

import models.Cat;

import models.enums.CatColor;

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.Assertions;

import org.mockito.Mockito;

import tools.CatsException;

import java.time.LocalDate;

import java.util.ArrayList;

import java.util.List;

import static org.mockito.Mockito.verify;

import static org.mockito.Mockito.when;

public class CatServiceTest {

private CatDao catDao;

private CatService catService;

@BeforeEach

public void setUp() {

catDao = Mockito.mock(CatDao.class);

catService = new CatService(catDao);

}

@Test

public void getCat() {

var mockedCat = new Cat("cat", LocalDate.now(), "breed", CatColor.BROWN);

when(catDao.getById(1)).thenReturn(mockedCat);

var cat = catService.getById(1);

verify(catDao).getById(1);

Assertions.assertEquals(mockedCat, cat);

}

@Test

public void updateCat() throws CatsException {

var mockedCat1 = new Cat("cat1", LocalDate.now(), "breed", CatColor.BROWN);

var mockedCat2 = new Cat("cat2", LocalDate.now(), "breed", CatColor.BLACK);

when(catDao.getById(1)).thenReturn(mockedCat1);

when(catDao.getById(2)).thenReturn(mockedCat2);

var cat1 = catService.getById(1);

Assertions.assertEquals(0, cat1.getFriends().size());

mockedCat1.addFriend(mockedCat2);

catService.save(cat1);

verify(catDao).save(cat1);

var cat2 = catService.getById(2);

Assertions.assertEquals(1, cat1.getFriends().size());

Assertions.assertEquals(1, cat2.getFriends().size());

}

@Test

public void findAllCats() {

var mockedCat1 = new Cat("cat", LocalDate.now(), "breed", CatColor.BROWN);

var mockedCat2 = new Cat("cat2", LocalDate.now(), "breed", CatColor.BLACK);

List<Cat> mockedCats = new ArrayList<>();

mockedCats.add(mockedCat1);

mockedCats.add(mockedCat2);

when(catDao.findAll()).thenReturn(mockedCats);

var catList = catService.findAll();

verify(catDao).findAll();

Assertions.assertEquals(mockedCats, catList);

}

@Test

public void deleteCat() {

when(catDao.findById(1)).thenReturn(null);

catService.deleteById(1);

verify(catDao).deleteById(1);

Assertions.assertNull(catService.getById(1));

}

}

Добавлен файл kotiki-java/user-service/src/test/java/services/OwnerServiceTest.java:

package services;

import dao.OwnerDao;

import models.Cat;

import models.Owner;

import models.enums.CatColor;

import org.junit.jupiter.api.Assertions;

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

import java.time.LocalDate;

import java.util.ArrayList;

import java.util.List;

import static org.mockito.Mockito.verify;

import static org.mockito.Mockito.when;

public class OwnerServiceTest {

private OwnerDao ownerDao;

private OwnerService ownerService;

@BeforeEach

public void setUp() {

ownerDao = Mockito.mock(OwnerDao.class);

ownerService = new OwnerService(ownerDao);

}

@Test

public void getOwner() {

var mockedOwner = new Owner("owner", LocalDate.now());

when(ownerDao.getById(1)).thenReturn(mockedOwner);

var owner = ownerService.getById(1);

verify(ownerDao).getById(1);

Assertions.assertEquals(mockedOwner, owner);

}

@Test

public void updateOwner() {

var mockedOwner = new Owner("owner", LocalDate.now());

var mockedCat = new Cat("cat2", LocalDate.now(), "breed", CatColor.BLACK);

when(ownerDao.getById(1)).thenReturn(mockedOwner);

var owner = ownerService.getById(1);

Assertions.assertEquals(0, owner.getCats().size());

mockedOwner.addCat(mockedCat);

ownerService.save(owner);

verify(ownerDao).save(owner);

Assertions.assertEquals(1, owner.getCats().size());

}

@Test

public void findAllOwners() {

var mockedOwner1 = new Owner("owner1", LocalDate.now());

var mockedOwner2 = new Owner("owner2", LocalDate.now());

List<Owner> mockedOwners = new ArrayList<>();

mockedOwners.add(mockedOwner1);

mockedOwners.add(mockedOwner2);

when(ownerDao.findAll()).thenReturn(mockedOwners);

var ownerList = ownerService.findAll();

verify(ownerDao).findAll();

Assertions.assertEquals(mockedOwners, ownerList);

}

@Test

public void deleteOwner() {

when(ownerDao.getById(1)).thenReturn(null);

ownerService.deleteById(1);

verify(ownerDao).deleteById(1);

Assertions.assertNull(ownerService.getById(1));

}

}