**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе №5

по дисциплине «**Технологии программирования**»

Автор: Юрпалов Сергей Николаевич

Факультет: ИТиП

Группа: М32051



Санкт-Петербург 2022

Описание:

Бизнес прочитал статью о том, что микросервисы это круто и попросил нас разбить программу на микросервисы.

Из созданного приложения выделяются три микросервиса:

Микросервис доступа к котикам

Микросервис доступа к владельцам

Микросервис с внешними интерфейсами.

Все они являются разными приложениями.

Все созданные ранее эндпоинты и авторизация переезжает на третий микросервис.

Общение между микросервисами происходит посредством RabbitMQ/Kafka (на выбор студента).

Сервисы доступа к котикам и доступа к владельцам могут либо быть подключены к одной БД, либо иметь разные БД. Во втором случае недопустимо делать один запрос на получение данных из двух БД, запроса должно быть два (по одному в каждую).

Внимание: недопустимо передавать через RabbitMQ/Kafka JPA сущности. Рекомендуется создать отдельные оберточные классы.

Рекомендуется выделить модуль с JPA сущностями в отдельный подключаемый модуль.

Исходный код:

Добавлен файл kotiki-java/api-access/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">

<modelVersion>4.0.0</modelVersion>

<groupId>org.wilfordaf</groupId>

<artifactId>api-access</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

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

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

<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.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>

<dependency>

<groupId>org.wilfordaf</groupId>

<artifactId>domain-model</artifactId>

<version>1.0-SNAPSHOT</version>

<scope>compile</scope>

</dependency>

<dependency>

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

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

<version>2.6.5</version>

</dependency>

<dependency>

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

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

<version>2.6.5</version>

</dependency>

<dependency>

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

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

<version>2.6.5</version>

</dependency>

<dependency>

<groupId>org.postgresql</groupId>

<artifactId>postgresql</artifactId>

<version>42.3.4</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>javax.xml.bind</groupId>

<artifactId>jaxb-api</artifactId>

<version>2.3.1</version>

</dependency>

</dependencies>

<properties>

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

<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/api-access/src/main/java/org/wilfordaf/CatsSpringApplication.java:

package org.wilfordaf;

import com.fasterxml.jackson.databind.ObjectMapper;

import com.fasterxml.jackson.databind.util.StdDateFormat;

import com.fasterxml.jackson.datatype.jsr310.JavaTimeModule;

import org.springframework.amqp.core.\*;

import org.springframework.amqp.rabbit.connection.ConnectionFactory;

import org.springframework.amqp.rabbit.core.RabbitTemplate;

import org.springframework.amqp.support.converter.Jackson2JsonMessageConverter;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.Bean;

import org.springframework.http.converter.json.Jackson2ObjectMapperBuilder;

@SpringBootApplication

public class CatsSpringApplication {

public static final String CAT\_EXCHANGE = "cat-exchange";

public static final String CAT\_SAVE\_QUEUE = "cat-save-queue";

public static final String CAT\_SAVE\_ROUTING\_KEY = "cat-save-routing-key";

public static final String CAT\_GET\_QUEUE = "cat-get-queue";

public static final String CAT\_GET\_ROUTING\_KEY = "cat-get-routing-key";

public static final String CAT\_FIND\_BY\_OWNER\_ID\_QUEUE = "cat-find-by-owner-id-queue";

public static final String CAT\_FIND\_BY\_OWNER\_ID\_ROUTING\_KEY = "cat-find-by-owner-id-routing-key";

public static final String CAT\_FIND\_ALL\_QUEUE = "cat-find-all-queue";

public static final String CAT\_FIND\_ALL\_ROUTING\_KEY = "cat-find-all-routing-key";

public static final String OWNER\_EXCHANGE = "owner-exchange";

public static final String OWNER\_SAVE\_QUEUE = "owner-save-queue";

public static final String OWNER\_SAVE\_ROUTING\_KEY = "owner-save-routing-key";

public static final String OWNER\_GET\_QUEUE = "owner-get-queue";

public static final String OWNER\_GET\_ROUTING\_KEY = "owner-get-routing-key";

public static final String OWNER\_FIND\_ALL\_QUEUE = "owner-find-all-queue";

public static final String OWNER\_FIND\_ALL\_ROUTING\_KEY = "owner-find-all-routing-key";

public static final String OWNER\_DELETE\_QUEUE = "owner-delete-queue";

public static final String OWNER\_DELETE\_ROUTING\_KEY = "owner-delete-routing-key";

public static final String USER\_EXCHANGE = "user-exchange";

public static final String USER\_GET\_BY\_USERNAME\_QUEUE = "user-get-by-username-queue";

public static final String USER\_GET\_BY\_USERNAME\_ROUTING\_KEY = "user-get-by-username-routing-key";

public static final String USER\_SAVE\_QUEUE = "user-save-queue";

public static final String USER\_SAVE\_ROUTING\_KEY = "user-save-routing-key";

public static final String USER\_DELETE\_QUEUE = "user-delete-queue";

public static final String USER\_DELETE\_ROUTING\_KEY = "user-delete-routing-key";

public static final String OWNER\_GET\_BY\_CAT\_ID\_QUEUE = "owner-get-by-cat-id-queue";

public static final String OWNER\_GET\_BY\_CAT\_ID\_ROUTING\_KEY = "owner-get-by-cat-id-routing-key";

public static final String CAT\_FIND\_BY\_BREED\_AND\_OWNER\_ID\_QUEUE = "cat-find-by-breed-and-owner-id-queue";

public static final String CAT\_FIND\_BY\_BREED\_AND\_OWNER\_ID\_ROUTING\_KEY = "cat-find-by-breed-and-owner-id-routing-key";

public static final String CAT\_FIND\_BY\_BREED\_QUEUE = "cat-find-by-breed-queue";

public static final String CAT\_FIND\_BY\_BREED\_ROUTING\_KEY = "cat-find-by-breed-routing-key";

public static final String CAT\_DELETE\_QUEUE = "cat-delete-queue";

public static final String CAT\_DELETE\_ROUTING\_KEY = "cat-delete-routing-key";

public static final String CAT\_FIND\_BY\_COLOR\_QUEUE = "cat-find-by-color-queue";

public static final String CAT\_FIND\_BY\_COLOR\_ROUTING\_KEY = "cat-find-by-color-routing-key";

public static final String CAT\_FIND\_BY\_COLOR\_AND\_OWNER\_ID\_QUEUE = "cat-find-by-color-and-owner-id-queue";

public static final String CAT\_FIND\_BY\_COLOR\_AND\_OWNER\_ID\_ROUTING\_KEY = "cat-find-by-color-and-owner-id-routing-key";

public static final String OWNER\_FIND\_BY\_NAME\_QUEUE = "owner-find-by-name-queue";

public static final String OWNER\_FIND\_BY\_NAME\_ROUTING\_KEY = "owner-find-by-name-routing-key";

public static final String OWNER\_FIND\_BY\_BIRTH\_DATE\_QUEUE = "owner-find-by-birth-date-queue";

public static final String OWNER\_FIND\_BY\_BIRTH\_DATE\_ROUTING\_KEY = "owner-find-by-birth-date-routing-key";

@Bean

public TopicExchange catExchange() {

return new TopicExchange(CAT\_EXCHANGE);

}

@Bean

public TopicExchange ownerExchange() {

return new TopicExchange(OWNER\_EXCHANGE);

}

@Bean

public Queue catSaveQueue() {

return new Queue(CAT\_SAVE\_QUEUE);

}

@Bean

public Binding catSaveBinding() {

return BindingBuilder.bind(catSaveQueue()).to(catExchange()).with(CAT\_SAVE\_ROUTING\_KEY);

}

@Bean

public Queue catFindByBreedQueue() {

return new Queue(CAT\_FIND\_BY\_BREED\_QUEUE);

}

@Bean

public Binding catFindByBreedBinding() {

return BindingBuilder.bind(catFindByBreedQueue()).to(catExchange()).with(CAT\_FIND\_BY\_BREED\_ROUTING\_KEY);

}

@Bean

public Queue catFindByBreedAndOwnerIdQueue() {

return new Queue(CAT\_FIND\_BY\_BREED\_AND\_OWNER\_ID\_QUEUE);

}

@Bean

public Binding catFindByBreedAndOwnerIdBinding() {

return BindingBuilder.bind(catFindByBreedAndOwnerIdQueue()).to(catExchange()).with(CAT\_FIND\_BY\_BREED\_AND\_OWNER\_ID\_ROUTING\_KEY);

}

@Bean

public Queue catFindByOwnerId() {

return new Queue(CAT\_FIND\_BY\_OWNER\_ID\_QUEUE);

}

@Bean

public Binding catFindByOwnerIdBinding() {

return BindingBuilder.bind(catFindByOwnerId()).to(catExchange()).with(CAT\_FIND\_BY\_OWNER\_ID\_ROUTING\_KEY);

}

@Bean

public Queue ownerSaveQueue() {

return new Queue(OWNER\_SAVE\_QUEUE);

}

@Bean

public Binding ownerSaveBinding() {

return BindingBuilder.bind(ownerSaveQueue()).to(ownerExchange()).with(OWNER\_SAVE\_ROUTING\_KEY);

}

@Bean

public Queue catGetQueue() {

return new Queue(CAT\_GET\_QUEUE);

}

@Bean

public Binding catGetBinding() {

return BindingBuilder.bind(catGetQueue()).to(catExchange()).with(CAT\_GET\_ROUTING\_KEY);

}

@Bean

public Queue ownerGetQueue() {

return new Queue(OWNER\_GET\_QUEUE);

}

@Bean

public Binding ownerGetBinding() {

return BindingBuilder.bind(ownerGetQueue()).to(ownerExchange()).with(OWNER\_GET\_ROUTING\_KEY);

}

@Bean

public Queue ownerFindAllQueue() {

return new Queue(OWNER\_FIND\_ALL\_QUEUE);

}

@Bean

public Binding ownerFindAllBinding() {

return BindingBuilder.bind(ownerFindAllQueue()).to(ownerExchange()).with(OWNER\_FIND\_ALL\_ROUTING\_KEY);

}

@Bean

public Queue ownerDeleteQueue() {

return new Queue(OWNER\_DELETE\_QUEUE);

}

@Bean

public Binding ownerDeleteBinding() {

return BindingBuilder.bind(ownerDeleteQueue()).to(ownerExchange()).with(OWNER\_DELETE\_ROUTING\_KEY);

}

@Bean

public TopicExchange userExchange() {

return new TopicExchange(USER\_EXCHANGE);

}

@Bean

public Queue userGetByUsernameQueue() {

return new Queue(USER\_GET\_BY\_USERNAME\_QUEUE);

}

@Bean

public Binding userGetByUsernameBinding() {

return BindingBuilder.bind(userGetByUsernameQueue()).to(userExchange()).with(USER\_GET\_BY\_USERNAME\_ROUTING\_KEY);

}

@Bean

public Queue userSaveQueue() {

return new Queue(USER\_SAVE\_QUEUE);

}

@Bean

public Binding userSaveBinding() {

return BindingBuilder.bind(userSaveQueue()).to(userExchange()).with(USER\_SAVE\_ROUTING\_KEY);

}

@Bean

public Queue userDeleteQueue() {

return new Queue(USER\_DELETE\_QUEUE);

}

@Bean

public Binding userDeleteBinding() {

return BindingBuilder.bind(userDeleteQueue()).to(userExchange()).with(USER\_DELETE\_ROUTING\_KEY);

}

@Bean

public Queue ownerGetByCatIdQueue() {

return new Queue(OWNER\_GET\_BY\_CAT\_ID\_QUEUE);

}

@Bean

public Binding ownerGetByCatIdBinding() {

return BindingBuilder.bind(ownerGetByCatIdQueue()).to(ownerExchange()).with(OWNER\_GET\_BY\_CAT\_ID\_ROUTING\_KEY);

}

@Bean

public Queue catFindAllQueue() {

return new Queue(CAT\_FIND\_ALL\_QUEUE);

}

@Bean

public Binding catFindAllBinding() {

return BindingBuilder.bind(catFindAllQueue()).to(catExchange()).with(CAT\_FIND\_ALL\_ROUTING\_KEY);

}

@Bean

public Queue catDeleteQueue() {

return new Queue(CAT\_DELETE\_QUEUE);

}

@Bean

public Binding catDeleteBinding() {

return BindingBuilder.bind(catDeleteQueue()).to(catExchange()).with(CAT\_DELETE\_ROUTING\_KEY);

}

@Bean

public Queue catFindByColorQueue() {

return new Queue(CAT\_FIND\_BY\_COLOR\_QUEUE);

}

@Bean

public Binding catFindByColorBinding() {

return BindingBuilder.bind(catFindByColorQueue()).to(catExchange()).with(CAT\_FIND\_BY\_COLOR\_ROUTING\_KEY);

}

@Bean

public Queue catFindByColorAndOwnerIdQueue() {

return new Queue(CAT\_FIND\_BY\_COLOR\_AND\_OWNER\_ID\_QUEUE);

}

@Bean

public Binding catFindByColorAndOwnerIdBinding() {

return BindingBuilder.bind(catFindByColorAndOwnerIdQueue()).to(catExchange()).with(CAT\_FIND\_BY\_COLOR\_AND\_OWNER\_ID\_ROUTING\_KEY);

}

@Bean

public Queue ownerFindByNameQueue() {

return new Queue(OWNER\_FIND\_BY\_NAME\_QUEUE);

}

@Bean

public Binding ownerFindByNameBinding() {

return BindingBuilder.bind(ownerFindByNameQueue()).to(ownerExchange()).with(OWNER\_FIND\_BY\_NAME\_ROUTING\_KEY);

}

@Bean

public Queue ownerFindByBirthDateQueue() {

return new Queue(OWNER\_FIND\_BY\_BIRTH\_DATE\_QUEUE);

}

@Bean

public Binding ownerFindByBirthDateBinding() {

return BindingBuilder.bind(ownerFindByBirthDateQueue()).to(ownerExchange()).with(OWNER\_FIND\_BY\_BIRTH\_DATE\_ROUTING\_KEY);

}

@Bean

public Jackson2JsonMessageConverter converter() {

ObjectMapper objectMapper = Jackson2ObjectMapperBuilder.json()

.modules(new JavaTimeModule())

.dateFormat(new StdDateFormat())

.timeZone("Europe/Moscow")

.build();

return new Jackson2JsonMessageConverter(objectMapper);

}

@Bean

public AmqpTemplate template(ConnectionFactory connectionFactory) {

RabbitTemplate rabbitTemplate = new RabbitTemplate(connectionFactory);

rabbitTemplate.setMessageConverter(converter());

return rabbitTemplate;

}

public static void main(String[] args) {

SpringApplication.run(CatsSpringApplication.class, args);

}

}

Изменен файл kotiki-java/user-service/src/main/java/startapp/cfg/SecurityConfig.java -> kotiki-java/api-access/src/main/java/org/wilfordaf/cfg/SecurityConfig.java:

-package startapp.cfg;

+package org.wilfordaf.cfg;

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

import org.springframework.context.annotation.Bean;

@@ -9,7 +9,7 @@

import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import org.springframework.security.crypto.password.PasswordEncoder;

-import services.UserService;

+import org.wilfordaf.services.UserService;

@EnableWebSecurity

@EnableGlobalMethodSecurity(prePostEnabled = true)

public DaoAuthenticationProvider authenticationProvider() {

authenticationProvider.setPasswordEncoder(passwordEncoder());

return authenticationProvider;

}

-}

+}

Изменен файл kotiki-java/user-service/src/main/java/controllers/CatController.java -> kotiki-java/api-access/src/main/java/org/wilfordaf/controllers/CatController.java:

-package controllers;

+package org.wilfordaf.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.security.access.prepost.PreAuthorize;

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

import org.springframework.web.server.ResponseStatusException;

-import services.CatService;

+import org.wilfordaf.dto.CatDto;

+import org.wilfordaf.enums.CatColor;

+import org.wilfordaf.services.CatService;

import javax.persistence.EntityNotFoundException;

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) {

+ public CatController(CatService catService) {

this.catService = catService;

- this.catToDtoMapper = catToDtoMapper;

- this.dtoToCatMapper = dtoToCatMapper;

}

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

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

try {

- Cat cat = catService.getById(id);

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

+ return ResponseEntity.ok(catService.getById(id));

} catch (EntityNotFoundException | NullPointerException e) {

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

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

@PostMapping("/save")

@PreAuthorize("hasRole('ADMIN')")

public void save(@RequestBody CatDto catDto) {

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

+ catService.save(catDto);

}

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

public void deleteById(@PathVariable int id) {

@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();

+ return catService.findAll();

}

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

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

- return catService.findByBirthDate(birthDate).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();

+ return catService.findByBreed(breed);

}

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

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

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

+ return catService.findByColor(color);

}

}

Изменен файл kotiki-java/user-service/src/main/java/controllers/OwnerController.java -> kotiki-java/api-access/src/main/java/org/wilfordaf/controllers/OwnerController.java:

-package controllers;

+package org.wilfordaf.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.security.access.prepost.PreAuthorize;

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

import org.springframework.web.server.ResponseStatusException;

-import services.OwnerService;

+import org.wilfordaf.dto.OwnerDto;

+import org.wilfordaf.services.OwnerService;

import javax.persistence.EntityNotFoundException;

import java.util.List;

@@ -20,21 +17,16 @@

public class OwnerController {

private final OwnerService ownerService;

- private final OwnerToDtoMapper ownerToDtoMapper;

- private final DtoToOwnerMapper dtoToOwnerMapper;

@Autowired

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

+ public OwnerController(OwnerService ownerService) {

this.ownerService = ownerService;

- this.ownerToDtoMapper = ownerToDtoMapper;

- this.dtoToOwnerMapper = dtoToOwnerMapper;

}

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

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

try {

- Owner owner = ownerService.getById(id);

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

+ return ResponseEntity.ok(ownerService.getById(id));

} catch (EntityNotFoundException | NullPointerException e) {

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

}

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

@PostMapping("/save")

@PreAuthorize("hasRole('ADMIN')")

public void save(@RequestBody OwnerDto ownerDto) {

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

+ ownerService.save(ownerDto);

}

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

public void deleteById(@PathVariable int id) {

@GetMapping ("/find-all")

@PreAuthorize("hasRole('ADMIN')")

public List<OwnerDto> findAll() {

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

+ return ownerService.findAll();

}

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

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

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

+ try {

+ return ownerService.findByName(name);

+ }

+ catch (NullPointerException e) {

+ throw new ResponseStatusException(HttpStatus.BAD\_REQUEST);

+ }

}

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

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

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

+ try {

+ return ownerService.findByBirthDate(birthDate);

+ }

+ catch (NullPointerException e) {

+ throw new ResponseStatusException(HttpStatus.BAD\_REQUEST);

+ }

}

}

+

Изменен файл kotiki-java/user-service/src/main/java/controllers/UserController.java -> kotiki-java/api-access/src/main/java/org/wilfordaf/controllers/UserController.java:

-package controllers;

+package org.wilfordaf.controllers;

-import dto.UserDto;

-import dto.mappers.DtoToUserMapper;

+import org.springframework.security.access.prepost.PreAuthorize;

+import org.wilfordaf.dto.UserDto;

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

import org.springframework.http.HttpStatus;

-import org.springframework.security.access.prepost.PreAuthorize;

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

import org.springframework.web.server.ResponseStatusException;

-import services.UserService;

+import org.wilfordaf.services.UserService;

@RestController

@RequestMapping("/users")

public class UserController {

private final UserService userService;

- private final DtoToUserMapper dtoToUserMapper;

@Autowired

- public UserController(UserService userService, DtoToUserMapper dtoToUserMapper) {

+ public UserController(UserService userService) {

this.userService = userService;

- this.dtoToUserMapper = dtoToUserMapper;

}

@PostMapping("/save")

@PreAuthorize("hasRole('ADMIN')")

public void save(@RequestBody UserDto user) {

- userService.save(dtoToUserMapper.convert(user));

+ userService.save(user);

}

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

Добавлен файл kotiki-java/api-access/src/main/java/org/wilfordaf/services/CatService.java:

package org.wilfordaf.services;

import org.springframework.amqp.rabbit.core.RabbitTemplate;

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

import org.springframework.core.ParameterizedTypeReference;

import org.springframework.stereotype.Service;

import org.wilfordaf.CatsSpringApplication;

import org.wilfordaf.dto.\*;

import org.wilfordaf.enums.Authority;

import org.wilfordaf.enums.CatColor;

import java.util.List;

@Service

@SuppressWarnings("unchecked")

public class CatService {

private final RabbitTemplate template;

private final UserService userService;

private final OwnerService ownerService;

@Autowired

public CatService(RabbitTemplate template, UserService userService, OwnerService ownerService) {

this.template = template;

this.userService = userService;

this.ownerService = ownerService;

}

public CatDto getById(int id) {

CatDto cat = template.convertSendAndReceiveAsType(

CatsSpringApplication.CAT\_EXCHANGE,

CatsSpringApplication.CAT\_GET\_ROUTING\_KEY,

id,

ParameterizedTypeReference.forType(CatDto.class));

UserDto user = userService.getAuthorizedUser();

if (user.getRole() == Authority.ROLE\_ADMIN) {

return cat;

}

if (ownerService.getByCatId(id).getId() == user.getOwner().getId()) {

return cat;

}

throw new NullPointerException();

}

public void save(CatDto catDto) {

UserDto user = userService.getAuthorizedUser();

if (user.getRole() == Authority.ROLE\_ADMIN) {

template.convertAndSend(CatsSpringApplication.CAT\_EXCHANGE, CatsSpringApplication.CAT\_SAVE\_ROUTING\_KEY, catDto);

return;

}

if (ownerService.getByCatId(catDto.getId()).getId() == user.getOwner().getId()) {

template.convertAndSend(CatsSpringApplication.CAT\_EXCHANGE, CatsSpringApplication.CAT\_SAVE\_ROUTING\_KEY, catDto);

}

}

public void deleteById(int id) {

template.convertAndSend(CatsSpringApplication.CAT\_EXCHANGE, CatsSpringApplication.CAT\_DELETE\_ROUTING\_KEY, id);

}

public List<CatDto> findAll() {

UserDto user = userService.getAuthorizedUser();

if (user.getRole() == Authority.ROLE\_ADMIN) {

return (List<CatDto>) template.convertSendAndReceive(

CatsSpringApplication.CAT\_EXCHANGE,

CatsSpringApplication.CAT\_FIND\_ALL\_ROUTING\_KEY,

"cat-find-all");

}

return (List<CatDto>) template.convertSendAndReceive(

CatsSpringApplication.CAT\_EXCHANGE,

CatsSpringApplication.CAT\_FIND\_BY\_OWNER\_ID\_ROUTING\_KEY,

user.getOwner().getId());

}

public List<CatDto> findByBreed(String breed) {

UserDto user = userService.getAuthorizedUser();

if (user.getRole() == Authority.ROLE\_ADMIN) {

return (List<CatDto>) template.convertSendAndReceive(

CatsSpringApplication.CAT\_EXCHANGE,

CatsSpringApplication.CAT\_FIND\_BY\_BREED\_ROUTING\_KEY,

breed);

}

return (List<CatDto>) template.convertSendAndReceive(

CatsSpringApplication.CAT\_EXCHANGE,

CatsSpringApplication.CAT\_FIND\_BY\_BREED\_AND\_OWNER\_ID\_ROUTING\_KEY,

new OwnerBreedDto(breed, user.getOwner().getId()));

}

public List<CatDto> findByColor(CatColor color) {

UserDto user = userService.getAuthorizedUser();

if (user.getRole() == Authority.ROLE\_ADMIN) {

return (List<CatDto>) template.convertSendAndReceive(

CatsSpringApplication.CAT\_EXCHANGE,

CatsSpringApplication.CAT\_FIND\_BY\_COLOR\_ROUTING\_KEY,

color);

}

return (List<CatDto>) template.convertSendAndReceive(

CatsSpringApplication.CAT\_EXCHANGE,

CatsSpringApplication.CAT\_FIND\_BY\_COLOR\_AND\_OWNER\_ID\_ROUTING\_KEY,

new OwnerColorDto(color, user.getOwner().getId()));

}

}

Добавлен файл kotiki-java/api-access/src/main/java/org/wilfordaf/services/OwnerService.java:

package org.wilfordaf.services;

import org.springframework.amqp.rabbit.core.RabbitTemplate;

import org.springframework.core.ParameterizedTypeReference;

import org.wilfordaf.CatsSpringApplication;

import org.wilfordaf.dto.OwnerDto;

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

import org.springframework.stereotype.Service;

import org.wilfordaf.dto.UserDto;

import org.wilfordaf.enums.Authority;

import java.time.LocalDate;

import java.util.List;

@Service

@SuppressWarnings("unchecked")

public class OwnerService {

private final RabbitTemplate template;

private final UserService userService;

@Autowired

public OwnerService(RabbitTemplate template, UserService userService) {

this.template = template;

this.userService = userService;

}

public OwnerDto getById(int id) {

UserDto user = userService.getAuthorizedUser();

if (user.getRole() == Authority.ROLE\_ADMIN) {

return template.convertSendAndReceiveAsType(

CatsSpringApplication.OWNER\_EXCHANGE,

CatsSpringApplication.OWNER\_GET\_ROUTING\_KEY,

id,

ParameterizedTypeReference.forType(OwnerDto.class));

}

OwnerDto owner = user.getOwner();

if (owner.getId() == id) {

return owner;

}

throw new NullPointerException();

}

public void save(OwnerDto ownerDto) {

template.convertAndSend(CatsSpringApplication.OWNER\_EXCHANGE, CatsSpringApplication.OWNER\_SAVE\_ROUTING\_KEY, ownerDto);

}

public void deleteById(int id) {

template.convertAndSend(CatsSpringApplication.OWNER\_EXCHANGE, CatsSpringApplication.OWNER\_DELETE\_ROUTING\_KEY, id);

}

public List<OwnerDto> findAll() {

UserDto user = userService.getAuthorizedUser();

if (user.getRole() == Authority.ROLE\_ADMIN) {

return (List<OwnerDto>) template.convertSendAndReceive(

CatsSpringApplication.OWNER\_EXCHANGE,

CatsSpringApplication.OWNER\_FIND\_ALL\_ROUTING\_KEY,

"owners-find-all");

}

return List.of(user.getOwner());

}

public List<OwnerDto> findByName(String name) {

UserDto user = userService.getAuthorizedUser();

if (user.getRole() == Authority.ROLE\_ADMIN) {

return (List<OwnerDto>) template.convertSendAndReceive(

CatsSpringApplication.OWNER\_EXCHANGE,

CatsSpringApplication.OWNER\_FIND\_BY\_NAME\_ROUTING\_KEY,

name);

}

if (user.getOwner().getName().equals(name)) {

return List.of(user.getOwner());

}

throw new NullPointerException();

}

public List<OwnerDto> findByBirthDate(String birthDate) {

LocalDate date = LocalDate.parse(birthDate);

UserDto user = userService.getAuthorizedUser();

if (user.getRole() == Authority.ROLE\_ADMIN) {

return (List<OwnerDto>) template.convertSendAndReceive(

CatsSpringApplication.OWNER\_EXCHANGE,

CatsSpringApplication.OWNER\_FIND\_BY\_BIRTH\_DATE\_ROUTING\_KEY,

date);

}

if (user.getOwner().getBirthDate() == date) {

return List.of(user.getOwner());

}

throw new NullPointerException();

}

public OwnerDto getByCatId(int id) {

return template.convertSendAndReceiveAsType(

CatsSpringApplication.OWNER\_EXCHANGE,

CatsSpringApplication.OWNER\_GET\_BY\_CAT\_ID\_ROUTING\_KEY,

id,

ParameterizedTypeReference.forType(OwnerDto.class));

}

}

Добавлен файл kotiki-java/api-access/src/main/java/org/wilfordaf/services/UserService.java:

package org.wilfordaf.services;

import org.springframework.amqp.rabbit.core.RabbitTemplate;

import org.springframework.core.ParameterizedTypeReference;

import org.springframework.security.core.authority.SimpleGrantedAuthority;

import org.springframework.security.core.context.SecurityContextHolder;

import org.springframework.security.core.userdetails.UserDetails;

import org.springframework.security.core.userdetails.UserDetailsService;

import org.springframework.security.core.userdetails.UsernameNotFoundException;

import org.wilfordaf.CatsSpringApplication;

import org.wilfordaf.dto.UserDto;

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

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class UserService implements UserDetailsService {

private final RabbitTemplate template;

@Autowired

public UserService(RabbitTemplate template) {

this.template = template;

}

@Override

public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

UserDto userDto = getByUsername(username);

if (userDto == null) {

throw new UsernameNotFoundException("User with username " + username + " was not found");

}

return new org.springframework.security.core.userdetails.User(

username,

userDto.getPassword(),

List.of(new SimpleGrantedAuthority(userDto.getRole().toString())));

}

public UserDto getAuthorizedUser() {

Object principal = SecurityContextHolder.getContext().getAuthentication().getPrincipal();

return getByUsername(principal instanceof UserDetails userDetails

? userDetails.getUsername() : principal.toString());

}

public UserDto getByUsername(String username) {

return template.convertSendAndReceiveAsType(

CatsSpringApplication.USER\_EXCHANGE,

CatsSpringApplication.USER\_GET\_BY\_USERNAME\_ROUTING\_KEY,

username,

ParameterizedTypeReference.forType(UserDto.class));

}

public void save(UserDto userDto) {

template.convertAndSend(CatsSpringApplication.USER\_EXCHANGE, CatsSpringApplication.USER\_SAVE\_ROUTING\_KEY, userDto);

}

public void deleteById(int id) {

template.convertAndSend(CatsSpringApplication.USER\_EXCHANGE, CatsSpringApplication.USER\_DELETE\_ROUTING\_KEY, id);

}

}

Добавлен файл kotiki-java/api-access/src/main/resources/application.properties:

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

spring.datasource.username=postgres

spring.datasource.password=123456

spring.jpa.hibernate.ddl-auto=update

spring.rabbitmq.host=localhost

spring.rabbitmq.username=guest

spring.rabbitmq.password=guest

spring.rabbitmq.port=5672

Изменен файл kotiki-java/user-service/src/test/java/services/CatServiceTest.java -> kotiki-java/api-access/src/test/java/org/wilfordaf/services/CatServiceTest.java:

-package services;

+/\*

+package org.wilfordaf.services;

-import dao.CatDao;

-import models.Cat;

-import models.enums.CatColor;

+import org.wilfordaf.dao.CatDao;

+import org.wilfordaf.enums.Authority;

+import org.wilfordaf.models.Cat;

+import org.wilfordaf.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 org.wilfordaf.models.User;

+import org.wilfordaf.tools.CatsException;

import java.time.LocalDate;

import java.util.ArrayList;

public class CatServiceTest {

private CatDao catDao;

private CatService catService;

- private UserService userService;

@BeforeEach

public void setUp() {

catDao = Mockito.mock(CatDao.class);

- userService = Mockito.mock(UserService.class);

+ var userService = Mockito.mock(UserService.class);

+ when(userService.getAuthorizedUser()).thenReturn(new User("123", "123", Authority.ROLE\_ADMIN, null));

catService = new CatService(catDao, userService);

}

public void deleteCat() {

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

}

}

+\*/

Изменен файл kotiki-java/user-service/src/test/java/services/OwnerServiceTest.java -> kotiki-java/api-access/src/test/java/org/wilfordaf/services/OwnerServiceTest.java:

-package services;

-

-import dao.OwnerDao;

-import models.Cat;

-import models.Owner;

-import models.enums.CatColor;

+/\*

+package org.wilfordaf.services;

+

+import org.wilfordaf.dao.OwnerDao;

+import org.wilfordaf.enums.Authority;

+import org.wilfordaf.models.Cat;

+import org.wilfordaf.models.Owner;

+import org.wilfordaf.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 org.wilfordaf.models.User;

import java.time.LocalDate;

import java.util.ArrayList;

public class OwnerServiceTest {

private OwnerDao ownerDao;

private OwnerService ownerService;

- private UserService userService;

@BeforeEach

public void setUp() {

ownerDao = Mockito.mock(OwnerDao.class);

- userService = Mockito.mock(UserService.class);

+ var userService = Mockito.mock(UserService.class);

+ when(userService.getAuthorizedUser()).thenReturn(new User("123", "123", Authority.ROLE\_ADMIN, null));

ownerService = new OwnerService(ownerDao, userService);

}

public void deleteOwner() {

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

}

}

+\*/

Добавлен файл kotiki-java/cat-access/pom.xml:

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

<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>org.wilfordaf</groupId>

<artifactId>cat-access</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<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>

<dependency>

<groupId>org.wilfordaf</groupId>

<artifactId>persistence</artifactId>

<version>1.0-SNAPSHOT</version>

<scope>compile</scope>

</dependency>

<dependency>

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

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

<version>2.6.5</version>

</dependency>

<dependency>

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

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

<version>2.6.5</version>

</dependency>

<dependency>

<groupId>org.postgresql</groupId>

<artifactId>postgresql</artifactId>

<version>42.3.4</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>javax.xml.bind</groupId>

<artifactId>jaxb-api</artifactId>

<version>2.3.1</version>

</dependency>

</dependencies>

<properties>

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

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

</properties>

</project>

Добавлен файл kotiki-java/cat-access/src/main/java/org/wilfordaf/CatApplication.java:

package org.wilfordaf;

import com.fasterxml.jackson.databind.ObjectMapper;

import com.fasterxml.jackson.databind.util.StdDateFormat;

import com.fasterxml.jackson.datatype.jsr310.JavaTimeModule;

import org.springframework.amqp.core.\*;

import org.springframework.amqp.support.converter.Jackson2JsonMessageConverter;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.Bean;

import org.springframework.http.converter.json.Jackson2ObjectMapperBuilder;

@SpringBootApplication

public class CatApplication {

public static final String CAT\_SAVE\_QUEUE = "cat-save-queue";

public static final String CAT\_GET\_QUEUE = "cat-get-queue";

public static final String CAT\_DELETE\_QUEUE = "cat-delete-queue";

public static final String CAT\_FIND\_BY\_OWNER\_ID\_QUEUE = "cat-find-by-owner-id-queue";

public static final String CAT\_FIND\_ALL\_QUEUE = "cat-find-all-queue";

public static final String CAT\_FIND\_BY\_BREED\_AND\_OWNER\_ID\_QUEUE = "cat-find-by-breed-and-owner-id-queue";

public static final String CAT\_FIND\_BY\_BREED\_QUEUE = "cat-find-by-breed-queue";

public static final String CAT\_FIND\_BY\_COLOR\_QUEUE = "cat-find-by-color-queue";

public static final String CAT\_FIND\_BY\_COLOR\_AND\_OWNER\_ID\_QUEUE = "cat-find-by-color-and-owner-id-queue";

@Bean

public Queue catSaveQueue() {

return new Queue(CAT\_SAVE\_QUEUE);

}

@Bean

public Queue catGetQueue() {

return new Queue(CAT\_GET\_QUEUE);

}

@Bean

public Queue catGetByOwnerIdQueue() {

return new Queue(CAT\_FIND\_BY\_OWNER\_ID\_QUEUE);

}

@Bean

public Queue catFindAllQueue() {

return new Queue(CAT\_FIND\_ALL\_QUEUE);

}

@Bean

public Queue catFindByBreedAndOwnerIdQueue() {

return new Queue(CAT\_FIND\_BY\_BREED\_AND\_OWNER\_ID\_QUEUE);

}

@Bean

public Queue catFindByBreedQueue() {

return new Queue(CAT\_FIND\_BY\_BREED\_QUEUE);

}

@Bean

public Queue catDeleteQueue() {

return new Queue(CAT\_DELETE\_QUEUE);

}

@Bean

public Queue catFindByColorQueue() {

return new Queue(CAT\_FIND\_BY\_COLOR\_QUEUE);

}

@Bean

public Queue catFindByColorAndOwnerIdQueue() {

return new Queue(CAT\_FIND\_BY\_COLOR\_AND\_OWNER\_ID\_QUEUE);

}

@Bean

public Jackson2JsonMessageConverter converter() {

ObjectMapper objectMapper = Jackson2ObjectMapperBuilder.json()

.modules(new JavaTimeModule())

.dateFormat(new StdDateFormat())

.timeZone("Europe/Moscow")

.build();

return new Jackson2JsonMessageConverter(objectMapper);

}

public static void main(String[] args) {

SpringApplication.run(CatApplication.class, args);

}

}

Добавлен файл kotiki-java/cat-access/src/main/java/org/wilfordaf/dao/CatDao.java:

package org.wilfordaf.dao;

import org.wilfordaf.models.Cat;

import org.wilfordaf.models.Owner;

import org.wilfordaf.enums.CatColor;

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

import java.util.List;

public interface CatDao extends JpaRepository<Cat, Integer> {

List<Cat> findByOwnerId(int id);

List<Cat> findByBreed(String breed);

List<Cat> findByBreedAndOwnerId(String breed, int id);

List<Cat> findByColor(CatColor color);

List<Cat> findByColorAndOwnerId(CatColor color, int id);

}

Добавлен файл kotiki-java/cat-access/src/main/java/org/wilfordaf/listeners/CatListener.java:

package org.wilfordaf.listeners;

import org.springframework.amqp.rabbit.annotation.RabbitListener;

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

import org.springframework.stereotype.Component;

import org.wilfordaf.CatApplication;

import org.wilfordaf.dto.CatDto;

import org.wilfordaf.dto.OwnerBreedDto;

import org.wilfordaf.dto.OwnerColorDto;

import org.wilfordaf.enums.CatColor;

import org.wilfordaf.mappers.CatToDtoMapper;

import org.wilfordaf.mappers.DtoToCatMapper;

import org.wilfordaf.services.CatService;

import javax.transaction.Transactional;

import java.util.ArrayList;

import java.util.List;

@Component

public class CatListener {

private final CatService catService;

private final DtoToCatMapper dtoToCatMapper;

private final CatToDtoMapper catToDtoMapper;

@Autowired

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

this.catService = catService;

this.dtoToCatMapper = dtoToCatMapper;

this.catToDtoMapper = catToDtoMapper;

}

@RabbitListener(queues = CatApplication.CAT\_SAVE\_QUEUE)

public void save(CatDto catDto) {

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

}

@RabbitListener(queues = CatApplication.CAT\_GET\_QUEUE)

@Transactional

public CatDto getById(int id) {

return catToDtoMapper.convert(catService.getById(id));

}

@RabbitListener(queues = CatApplication.CAT\_FIND\_BY\_OWNER\_ID\_QUEUE)

public List<CatDto> findOwnerId(int id) {

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

}

@RabbitListener(queues = CatApplication.CAT\_FIND\_ALL\_QUEUE)

public List<CatDto> findAll() {

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

}

@RabbitListener(queues = CatApplication.CAT\_FIND\_BY\_BREED\_AND\_OWNER\_ID\_QUEUE)

public List<CatDto> findByBreedAndOwner(OwnerBreedDto ownerBreedDto) {

return catService

.findByBreedAndOwnerId(ownerBreedDto.getBreed(), ownerBreedDto.getOwnerId())

.stream()

.map(catToDtoMapper::convert)

.toList();

}

@RabbitListener(queues = CatApplication.CAT\_FIND\_BY\_BREED\_QUEUE)

public List<CatDto> findByBreed(String breed) {

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

}

@RabbitListener(queues = CatApplication.CAT\_DELETE\_QUEUE)

@Transactional

public void deleteById(int id) {

catService.deleteById(id);

}

@RabbitListener(queues = CatApplication.CAT\_FIND\_BY\_COLOR\_QUEUE)

public List<CatDto> findByColor(CatColor color) {

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

}

@RabbitListener(queues = CatApplication.CAT\_FIND\_BY\_COLOR\_AND\_OWNER\_ID\_QUEUE)

public List<CatDto> findByColorAndOwnerId(OwnerColorDto ownerColorDto) {

return catService.findByColorAndOwnerId(ownerColorDto.getColor(), ownerColorDto.getOwnerId()).stream().map(catToDtoMapper::convert).toList();

}

}

Изменен файл kotiki-java/user-service/src/main/java/dto/mappers/CatToDtoMapper.java -> kotiki-java/cat-access/src/main/java/org/wilfordaf/mappers/CatToDtoMapper.java:

-package dto.mappers;

+package org.wilfordaf.mappers;

-import dto.CatDto;

-import models.Cat;

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

+import org.wilfordaf.dto.CatDto;

+import org.wilfordaf.models.Cat;

import org.springframework.stereotype.Component;

import java.util.stream.Collectors;

@@ -10,13 +9,6 @@

@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());

public CatDto convert(Cat cat) {

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;

}

Изменен файл kotiki-java/user-service/src/main/java/dto/mappers/DtoToCatMapper.java -> kotiki-java/cat-access/src/main/java/org/wilfordaf/mappers/DtoToCatMapper.java:

-package dto.mappers;

+package org.wilfordaf.mappers;

-import dto.CatDto;

-import models.Cat;

+import org.wilfordaf.dto.CatDto;

+import org.wilfordaf.models.Cat;

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

import org.springframework.stereotype.Component;

-import services.CatService;

+import org.wilfordaf.services.CatService;

import java.util.HashSet;

import java.util.Set;

@@ -14,12 +14,10 @@

public class DtoToCatMapper {

private final CatService catService;

- private final DtoToOwnerMapper dtoToOwnerMapper;

@Autowired

- public DtoToCatMapper(CatService catService, DtoToOwnerMapper dtoToOwnerMapper) {

+ public DtoToCatMapper(CatService catService) {

this.catService = catService;

- this.dtoToOwnerMapper = dtoToOwnerMapper;

}

public Cat convert(CatDto catDto) {

public Cat convert(CatDto catDto) {

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());

Добавлен файл kotiki-java/cat-access/src/main/java/org/wilfordaf/services/CatService.java:

package org.wilfordaf.services;

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

import org.springframework.stereotype.Service;

import org.wilfordaf.dao.CatDao;

import org.wilfordaf.enums.CatColor;

import org.wilfordaf.models.Cat;

import org.wilfordaf.models.Owner;

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> findByBreed(String breed) {

return catDao.findByBreed(breed);

}

public List<Cat> findByBreedAndOwnerId(String breed, int id) {

return catDao.findByBreedAndOwnerId(breed, id);

}

public List<Cat> findByColor(CatColor color) {

return catDao.findByColor(color);

}

public List<Cat> findByColorAndOwnerId(CatColor color, int id) {

return catDao.findByColorAndOwnerId(color, id);

}

public List<Cat> findByOwnerId(int id) {

return catDao.findByOwnerId(id);

}

}

Добавлен файл kotiki-java/cat-access/src/main/resources/application.properties:

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

spring.datasource.username=postgres

spring.datasource.password=123456

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.rabbitmq.host=localhost

spring.rabbitmq.username=guest

spring.rabbitmq.password=guest

spring.rabbitmq.port=5672

server.port=8082

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

Добавлен файл kotiki-java/domain-model/pom.xml:

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

<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>org.wilfordaf</groupId>

<artifactId>domain-model</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

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

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

</properties>

</project>

Изменен файл kotiki-java/user-service/src/main/java/dto/CatDto.java -> kotiki-java/domain-model/src/main/java/org/wilfordaf/dto/CatDto.java:

-package dto;

+package org.wilfordaf.dto;

-import models.enums.CatColor;

+import org.wilfordaf.enums.CatColor;

import java.time.LocalDate;

import java.util.\*;

public class CatDto {

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) {

+ public CatDto(int id, String name, LocalDate birthDate, String breed, CatColor color, 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 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;

}

Добавлен файл kotiki-java/domain-model/src/main/java/org/wilfordaf/dto/OwnerBreedDto.java:

package org.wilfordaf.dto;

public class OwnerBreedDto {

private String breed;

private int ownerId;

public OwnerBreedDto() {

}

public OwnerBreedDto(String breed, int ownerId) {

this.breed = breed;

this.ownerId = ownerId;

}

public String getBreed() {

return breed;

}

public void setBreed(String breed) {

this.breed = breed;

}

public int getOwnerId() {

return ownerId;

}

public void setOwnerId(int ownerId) {

this.ownerId = ownerId;

}

}

Добавлен файл kotiki-java/domain-model/src/main/java/org/wilfordaf/dto/OwnerColorDto.java:

package org.wilfordaf.dto;

import org.wilfordaf.enums.CatColor;

public class OwnerColorDto {

private CatColor color;

private int ownerId;

public OwnerColorDto() {

}

public OwnerColorDto(CatColor color, int ownerId) {

this.color = color;

this.ownerId = ownerId;

}

public CatColor getColor() {

return color;

}

public void setColor(CatColor color) {

this.color = color;

}

public int getOwnerId() {

return ownerId;

}

public void setOwnerId(int ownerId) {

this.ownerId = ownerId;

}

}

Изменен файл kotiki-java/user-service/src/main/java/dto/OwnerDto.java -> kotiki-java/domain-model/src/main/java/org/wilfordaf/dto/OwnerDto.java:

-package dto;

+package org.wilfordaf.dto;

import java.time.LocalDate;

-import java.util.HashSet;

-import java.util.Set;

public class OwnerDto {

public class OwnerDto {

private LocalDate birthDate;

- private Set<Integer> cats;

-

public OwnerDto() {

- cats = new HashSet<>();

}

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

+ public OwnerDto(int id, String name, LocalDate birthDate) {

this.id = id;

this.name = name;

this.birthDate = birthDate;

- this.cats = cats;

}

public int getId() {

public LocalDate getBirthDate() {

public void setBirthDate(LocalDate birthDate) {

this.birthDate = birthDate;

}

-

- public Set<Integer> getCats() {

- return cats;

- }

-

- public void setCats(Set<Integer> cats) {

- this.cats = cats;

- }

}

Изменен файл kotiki-java/user-service/src/main/java/dto/UserDto.java -> kotiki-java/domain-model/src/main/java/org/wilfordaf/dto/UserDto.java:

-package dto;

+package org.wilfordaf.dto;

-import models.enums.Authority;

+import org.wilfordaf.enums.Authority;

public class UserDto {

Изменен файл kotiki-java/database-access/src/main/java/models/enums/Authority.java -> kotiki-java/domain-model/src/main/java/org/wilfordaf/enums/Authority.java:

-package models.enums;

+package org.wilfordaf.enums;

public enum Authority {

Изменен файл kotiki-java/database-access/src/main/java/models/enums/CatColor.java -> kotiki-java/domain-model/src/main/java/org/wilfordaf/enums/CatColor.java:

-package models.enums;

+package org.wilfordaf.enums;

public enum CatColor {

Изменен файл kotiki-java/database-access/src/main/java/tools/CatsException.java -> kotiki-java/domain-model/src/main/java/org/wilfordaf/tools/CatsException.java:

-package tools;

+package org.wilfordaf.tools;

public class CatsException extends Exception {

Добавлен файл kotiki-java/owner-access/pom.xml:

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

<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>org.wilfordaf</groupId>

<artifactId>owner-access</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<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>

<dependency>

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

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

<version>2.6.5</version>

</dependency>

<dependency>

<groupId>org.wilfordaf</groupId>

<artifactId>persistence</artifactId>

<version>1.0-SNAPSHOT</version>

<scope>compile</scope>

</dependency>

<dependency>

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

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

<version>2.6.5</version>

</dependency>

<dependency>

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

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

<version>2.6.5</version>

</dependency>

<dependency>

<groupId>org.postgresql</groupId>

<artifactId>postgresql</artifactId>

<version>42.3.4</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>javax.xml.bind</groupId>

<artifactId>jaxb-api</artifactId>

<version>2.3.1</version>

</dependency>

</dependencies>

<properties>

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

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

</properties>

</project>

Добавлен файл kotiki-java/owner-access/src/main/java/org/wilfordaf/OwnerApplication.java:

package org.wilfordaf;

import com.fasterxml.jackson.databind.ObjectMapper;

import com.fasterxml.jackson.databind.util.StdDateFormat;

import com.fasterxml.jackson.datatype.jsr310.JavaTimeModule;

import org.springframework.amqp.core.Queue;

import org.springframework.amqp.support.converter.Jackson2JsonMessageConverter;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.Bean;

import org.springframework.http.converter.json.Jackson2ObjectMapperBuilder;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import org.springframework.security.crypto.password.PasswordEncoder;

@SpringBootApplication

public class OwnerApplication {

public static final String OWNER\_GET\_QUEUE = "owner-get-queue";

public static final String OWNER\_SAVE\_QUEUE = "owner-save-queue";

public static final String OWNER\_DELETE\_QUEUE = "owner-delete-queue";

public static final String OWNER\_FIND\_ALL\_QUEUE = "owner-find-all-queue";

public static final String OWNER\_GET\_BY\_CAT\_ID\_QUEUE = "owner-get-by-cat-id-queue";

public static final String OWNER\_FIND\_BY\_NAME\_QUEUE = "owner-find-by-name-queue";

public static final String OWNER\_FIND\_BY\_BIRTH\_DATE\_QUEUE = "owner-find-by-birth-date-queue";

public static final String USER\_GET\_BY\_USERNAME\_QUEUE = "user-get-by-username-queue";

public static final String USER\_SAVE\_QUEUE = "user-save-queue";

public static final String USER\_DELETE\_QUEUE = "user-delete-queue";

@Bean

public Queue ownerGetQueue() {

return new Queue(OWNER\_GET\_QUEUE);

}

@Bean

public Queue ownerSaveQueue() {

return new Queue(OWNER\_SAVE\_QUEUE);

}

@Bean

public Queue ownerDeleteQueue() {

return new Queue(OWNER\_DELETE\_QUEUE);

}

@Bean

public Queue ownerFindAllQueue() {

return new Queue(OWNER\_FIND\_ALL\_QUEUE);

}

@Bean

public Queue ownerGetByCatIdQueue() {

return new Queue(OWNER\_GET\_BY\_CAT\_ID\_QUEUE);

}

@Bean

public Queue ownerFindByNameQueue() {

return new Queue(OWNER\_FIND\_BY\_NAME\_QUEUE);

}

@Bean

public Queue ownerFindByBirthDateQueue() {

return new Queue(OWNER\_FIND\_BY\_BIRTH\_DATE\_QUEUE);

}

@Bean

public Queue userGetByUsernameQueue() {

return new Queue(USER\_GET\_BY\_USERNAME\_QUEUE);

}

@Bean

public Queue userSaveQueue() {

return new Queue(USER\_SAVE\_QUEUE);

}

@Bean

public Queue userDeleteQueue() {

return new Queue(USER\_DELETE\_QUEUE);

}

@Bean

public Jackson2JsonMessageConverter converter() {

ObjectMapper objectMapper = Jackson2ObjectMapperBuilder.json()

.modules(new JavaTimeModule())

.dateFormat(new StdDateFormat())

.timeZone("Europe/Moscow")

.build();

return new Jackson2JsonMessageConverter(objectMapper);

}

@Bean

public PasswordEncoder passwordEncoder() {

return new BCryptPasswordEncoder();

}

public static void main(String[] args) {

SpringApplication.run(OwnerApplication.class, args);

}

}

Изменен файл kotiki-java/database-access/src/main/java/dao/OwnerDao.java -> kotiki-java/owner-access/src/main/java/org/wilfordaf/dao/OwnerDao.java:

-package dao;

+package org.wilfordaf.dao;

-import models.Owner;

+import org.wilfordaf.models.Owner;

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

import java.time.LocalDate;

Изменен файл kotiki-java/database-access/src/main/java/dao/UserDao.java -> kotiki-java/owner-access/src/main/java/org/wilfordaf/dao/UserDao.java:

-package dao;

+package org.wilfordaf.dao;

-import models.User;

+import org.wilfordaf.models.User;

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

public interface UserDao extends JpaRepository<User, Integer> {

Добавлен файл kotiki-java/owner-access/src/main/java/org/wilfordaf/listeners/OwnerListener.java:

package org.wilfordaf.listeners;

import org.springframework.amqp.rabbit.annotation.RabbitListener;

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

import org.springframework.stereotype.Component;

import org.wilfordaf.OwnerApplication;

import org.wilfordaf.dto.OwnerDto;

import org.wilfordaf.mappers.DtoToOwnerMapper;

import org.wilfordaf.mappers.OwnerToDtoMapper;

import org.wilfordaf.services.OwnerService;

import javax.transaction.Transactional;

import java.time.LocalDate;

import java.util.List;

@Component

public class OwnerListener {

private final OwnerService ownerService;

private final DtoToOwnerMapper dtoToOwnerMapper;

private final OwnerToDtoMapper ownerToDtoMapper;

@Autowired

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

this.ownerService = ownerService;

this.dtoToOwnerMapper = dtoToOwnerMapper;

this.ownerToDtoMapper = ownerToDtoMapper;

}

@RabbitListener(queues = OwnerApplication.OWNER\_SAVE\_QUEUE)

public void save(OwnerDto ownerDto) {

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

}

@RabbitListener(queues = OwnerApplication.OWNER\_GET\_QUEUE)

@Transactional

public OwnerDto getById(int id) {

return ownerToDtoMapper.convert(ownerService.getById(id));

}

@RabbitListener(queues = OwnerApplication.OWNER\_GET\_BY\_CAT\_ID\_QUEUE)

@Transactional

public OwnerDto getByCatId(int id) {

return ownerToDtoMapper.convert(ownerService.getByCatId(id));

}

@RabbitListener(queues = OwnerApplication.OWNER\_FIND\_ALL\_QUEUE)

public List<OwnerDto> findAll(String ignoredMessage) {

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

}

@RabbitListener(queues = OwnerApplication.OWNER\_DELETE\_QUEUE)

@Transactional

public void deleteById(int id) {

ownerService.deleteById(id);

}

@RabbitListener(queues = OwnerApplication.OWNER\_FIND\_BY\_NAME\_QUEUE)

public List<OwnerDto> findByName(String name) {

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

}

@RabbitListener(queues = OwnerApplication.OWNER\_FIND\_BY\_NAME\_QUEUE)

public List<OwnerDto> findByBirthDate(String birthDate) {

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

}

}

Добавлен файл kotiki-java/owner-access/src/main/java/org/wilfordaf/listeners/UserListener.java:

package org.wilfordaf.listeners;

import org.springframework.amqp.rabbit.annotation.RabbitListener;

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

import org.springframework.stereotype.Component;

import org.wilfordaf.OwnerApplication;

import org.wilfordaf.dto.UserDto;

import org.wilfordaf.mappers.DtoToUserMapper;

import org.wilfordaf.mappers.UserToDtoMapper;

import org.wilfordaf.services.UserService;

import javax.transaction.Transactional;

@Component

public class UserListener {

private final UserService userService;

private final DtoToUserMapper dtoToUserMapper;

private final UserToDtoMapper userToDtoMapper;

@Autowired

public UserListener(UserService userService, DtoToUserMapper dtoToUserMapper, UserToDtoMapper userToDtoMapper) {

this.userService = userService;

this.dtoToUserMapper = dtoToUserMapper;

this.userToDtoMapper = userToDtoMapper;

}

@RabbitListener(queues = OwnerApplication.USER\_GET\_BY\_USERNAME\_QUEUE)

@Transactional

public UserDto getByUsername(String username) {

return userToDtoMapper.convert(userService.getByUsername(username));

}

@RabbitListener(queues = OwnerApplication.USER\_SAVE\_QUEUE)

public void save(UserDto userDto) {

userService.save(dtoToUserMapper.convert(userDto));

}

@RabbitListener(queues = OwnerApplication.USER\_DELETE\_QUEUE)

@Transactional

public void deleteById(int id) {

userService.deleteById(id);

}

}

Добавлен файл kotiki-java/owner-access/src/main/java/org/wilfordaf/mappers/DtoToOwnerMapper.java:

package org.wilfordaf.mappers;

import org.wilfordaf.dto.OwnerDto;

import org.wilfordaf.models.Owner;

import org.springframework.stereotype.Component;

@Component

public class DtoToOwnerMapper {

public Owner convert(OwnerDto ownerDto) {

Owner owner = new Owner();

owner.setId(ownerDto.getId());

owner.setName(ownerDto.getName());

owner.setBirthDate(ownerDto.getBirthDate());

return owner;

}

}

Изменен файл kotiki-java/user-service/src/main/java/dto/mappers/DtoToUserMapper.java -> kotiki-java/owner-access/src/main/java/org/wilfordaf/mappers/DtoToUserMapper.java:

-package dto.mappers;

+package org.wilfordaf.mappers;

-import dto.UserDto;

-import models.User;

+import org.wilfordaf.dto.UserDto;

+import org.wilfordaf.models.User;

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

import org.springframework.security.crypto.password.PasswordEncoder;

import org.springframework.stereotype.Component;

Изменен файл kotiki-java/user-service/src/main/java/dto/mappers/OwnerToDtoMapper.java -> kotiki-java/owner-access/src/main/java/org/wilfordaf/mappers/OwnerToDtoMapper.java:

-package dto.mappers;

+package org.wilfordaf.mappers;

-import dto.OwnerDto;

-import models.Cat;

-import models.Owner;

+import org.wilfordaf.dto.OwnerDto;

+import org.wilfordaf.models.Owner;

import org.springframework.stereotype.Component;

-import java.util.stream.Collectors;

-

@Component

public class OwnerToDtoMapper {

public OwnerDto convert(Owner owner) {

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/owner-access/src/main/java/org/wilfordaf/mappers/UserToDtoMapper.java:

package org.wilfordaf.mappers;

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

import org.springframework.stereotype.Component;

import org.wilfordaf.dto.UserDto;

import org.wilfordaf.models.User;

@Component

public class UserToDtoMapper {

private final OwnerToDtoMapper ownerToDtoMapper;

@Autowired

public UserToDtoMapper(OwnerToDtoMapper ownerToDtoMapper) {

this.ownerToDtoMapper = ownerToDtoMapper;

}

public UserDto convert(User user) {

UserDto userDto = new UserDto();

userDto.setId(user.getId());

userDto.setUsername(user.getUsername());

userDto.setPassword(user.getPassword());

userDto.setRole(user.getRole());

userDto.setOwner(ownerToDtoMapper.convert(user.getOwner()));

return userDto;

}

}

Добавлен файл kotiki-java/owner-access/src/main/java/org/wilfordaf/services/OwnerService.java:

package org.wilfordaf.services;

import org.wilfordaf.dao.OwnerDao;

import org.wilfordaf.models.Owner;

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

import org.springframework.stereotype.Service;

import java.time.LocalDate;

import java.util.List;

import java.util.stream.Collectors;

@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(String birthDate) {

LocalDate date = LocalDate.parse(birthDate);

return ownerDao.findByBirthDate(date);

}

public Owner getByCatId(int id) {

return ownerDao.findAll().stream().filter(o -> o.getCats().stream().

filter(c -> c.getId() == id).collect(Collectors.toSet()).size() > 0).

findFirst().orElse(null);

}

}

Добавлен файл kotiki-java/owner-access/src/main/java/org/wilfordaf/services/UserService.java:

package org.wilfordaf.services;

import org.wilfordaf.dao.UserDao;

import org.wilfordaf.models.User;

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

import org.springframework.stereotype.Service;

@Service

public class UserService {

private final UserDao userDao;

@Autowired

public UserService(UserDao userDao) {

this.userDao = userDao;

}

public User getByUsername(String username) {

return userDao.getByUsername(username);

}

public void save(User user) {

userDao.save(user);

}

public void deleteById(int id) {

userDao.deleteById(id);

}

}

Добавлен файл kotiki-java/owner-access/src/main/resources/application.properties:

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

spring.datasource.username=postgres

spring.datasource.password=123456

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.rabbitmq.host=localhost

spring.rabbitmq.username=guest

spring.rabbitmq.password=guest

spring.rabbitmq.port=5672

server.port=8081

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

-<?xml version="1.0"?>

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

<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>database-access</artifactId>

+ <groupId>org.wilfordaf</groupId>

+ <artifactId>persistence</artifactId>

+ <version>1.0-SNAPSHOT</version>

<properties>

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

@@ -17,16 +14,16 @@

</properties>

<dependencies>

- <dependency>

- <groupId>org.postgresql</groupId>

- <artifactId>postgresql</artifactId>

- <version>42.3.3</version>

- </dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.6.7.Final</version>

</dependency>

+ <dependency>

+ <groupId>org.wilfordaf</groupId>

+ <artifactId>domain-model</artifactId>

+ <version>1.0-SNAPSHOT</version>

+ <scope>compile</scope>

+ </dependency>

</dependencies>

-

</project>

Изменен файл kotiki-java/database-access/src/main/java/models/Cat.java -> kotiki-java/persistence/src/main/java/org/wilfordaf/models/Cat.java:

-package models;

+package org.wilfordaf.models;

-import models.enums.CatColor;

-import tools.CatsException;

+import org.wilfordaf.enums.CatColor;

+import org.wilfordaf.tools.CatsException;

import javax.persistence.\*;

import java.time.LocalDate;

public class Cat {

@Column(name = "color")

private CatColor color;

- @ManyToOne(fetch = FetchType.LAZY)

+ @ManyToOne

@JoinTable(

name = "owner\_cats",

joinColumns = @JoinColumn(name = "cat\_id"),

public class Cat {

)

private Owner owner;

- @ManyToMany(cascade = {CascadeType.PERSIST, CascadeType.MERGE, CascadeType.REFRESH})

+ @ManyToMany(

+ fetch = FetchType.EAGER,

+ cascade = {CascadeType.PERSIST, CascadeType.MERGE, CascadeType.REFRESH}

+ )

@JoinTable(name="cat\_friends",

joinColumns=@JoinColumn(name="cat\_id"),

inverseJoinColumns=@JoinColumn(name="friend\_id")

)

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

- @ManyToMany(mappedBy = "friends", cascade = {CascadeType.PERSIST, CascadeType.MERGE, CascadeType.REFRESH})

+ @ManyToMany(

+ mappedBy = "friends",

+ fetch = FetchType.EAGER,

+ cascade = {CascadeType.PERSIST, CascadeType.MERGE, CascadeType.REFRESH}

+ )

private Set<Cat> friendOf = new HashSet<>();

public Cat() {}

Изменен файл kotiki-java/database-access/src/main/java/models/Owner.java -> kotiki-java/persistence/src/main/java/org/wilfordaf/models/Owner.java:

-package models;

+package org.wilfordaf.models;

import javax.persistence.\*;

import java.time.LocalDate;

public class Owner {

@Column(name = "birth\_date", nullable = false)

private LocalDate birthDate;

- @OneToMany(mappedBy = "owner", cascade = CascadeType.ALL, fetch = FetchType.LAZY)

+ @OneToMany(mappedBy = "owner", cascade = CascadeType.ALL)

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

- @OneToOne(mappedBy = "owner", cascade = CascadeType.MERGE)

+ @OneToOne(mappedBy = "owner", cascade = {CascadeType.MERGE, CascadeType.REFRESH, CascadeType.REMOVE})

private User user;

public Owner() {}

Изменен файл kotiki-java/database-access/src/main/java/models/User.java -> kotiki-java/persistence/src/main/java/org/wilfordaf/models/User.java:

-package models;

+package org.wilfordaf.models;

-import models.enums.Authority;

+import org.wilfordaf.enums.Authority;

import javax.persistence.\*;

public class User {

@Column(name = "role", nullable = false)

private Authority role;

- @OneToOne(cascade = CascadeType.MERGE)

+ @OneToOne(cascade = {CascadeType.MERGE, CascadeType.REFRESH, CascadeType.REMOVE})

@JoinColumn(name = "owner\_id")

private Owner owner;

Изменен файл kotiki-java/pom.xml:

<artifactId>java-kotiki</artifactId>

<packaging>pom</packaging>

<version>1.0-SNAPSHOT</version>

+

<modules>

- <module>database-access</module>

- <module>user-service</module>

+ <module>api-access</module>

+ <module>cat-access</module>

+ <module>owner-access</module>

+ <module>persistence</module>

+ <module>domain-model</module>

</modules>

- <dependencies>

- <dependency>

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

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

- <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.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>

- <dependency>

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

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

- <version>2.6.5</version>

- </dependency>

- </dependencies>

<properties>

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

<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>

Удален файл kotiki-java/user-service/pom.xml

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

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

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

Удален файл kotiki-java/user-service/src/main/java/services/UserService.java

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

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