**Отчёт по курсовой работе по разработке REST-сервиса “Пристройство домашних животных”**

**Выполнила: Рзвякова В. В.**

Сначала я разработала базу данных: Pets (ID, AmountofPets, Name, Type\_ofPets) Family (ID, FIO, Address, Pets\_ID) Ivent(ID, Data, Family\_ID, Pets\_ID) Family 1-N Pets Ivent 1-N Pets.

Также я написала примерный код для разработки сервиса

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

import java.util.List;

@RestController

@RequestMapping("/api")

public class PetAdoptionController {

private final PetRepository petRepository;

private final FamilyRepository familyRepository;

private final IventRepository iventRepository;

public PetAdoptionController(PetRepository petRepository, FamilyRepository familyRepository, IventRepository iventRepository) {

this.petRepository = petRepository;

this.familyRepository = familyRepository;

this.iventRepository = iventRepository;

}

// Получение списка всех домашних животных

@GetMapping("/pets")

public List<Pet> getAllPets() {

return petRepository.findAll();

}

// Получение информации о семье по ID

@GetMapping("/families/{familyId}")

public Family getFamilyById(@PathVariable Long familyId) {

return familyRepository.findById(familyId)

.orElseThrow(() -> new ResourceNotFoundException("Family not found with ID: " + familyId));

}

// Создание новой семьи

@PostMapping("/families")

public Family createFamily(@RequestBody Family family) {

return familyRepository.save(family);

}

// Создание нового события

@PostMapping("/events")

public Ivent createIvent(@RequestBody Ivent ivent) {

return iventRepository.save(ivent);

}

// Принятие домашнего животного в семью

@PostMapping("/families/{familyId}/pets/{petId}")

public Family adoptPet(@PathVariable Long familyId, @PathVariable Long petId) {

Family family = familyRepository.findById(familyId)

.orElseThrow(() -> new ResourceNotFoundException("Family not found with ID: " + familyId));

Pet pet = petRepository.findById(petId)

.orElseThrow(() -> new ResourceNotFoundException("Pet not found with ID: " + petId));

family.getPets().add(pet);

return familyRepository.save(family);

}

}

Здесь я использовала Spring Boot для создания REST-сервиса. Также необходимо создать соответствующие модели (Pet, Family, Ivent) и репозитории (PetRepository, FamilyRepository, IventRepository) для взаимодействия с базой данных.

Создание модели на примере Pets:

1. Создание класса модели:

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Pet {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String type;

private int amount;

// Геттеры и сеттеры

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getType() {

return type;

}

public void setType(String type) {

this.type = type;

}

public int getAmount() {

return amount;

}

public void setAmount(int amount) {

this.amount = amount;

}

}

1. Добавляем репозиторий для Pets:

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

public interface PetRepository extends JpaRepository<Pet, Long> {

// Дополнительные методы, если необходимо

}

1. Используем модель в контроллере:

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

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

import java.util.List;

@RestController

@RequestMapping("/api/pets")

public class PetController {

private final PetRepository petRepository;

@Autowired

public PetController(PetRepository petRepository) {

this.petRepository = petRepository;

}

@GetMapping

public List<Pet> getAllPets() {

return petRepository.findAll();

}

@PostMapping

public Pet createPet(@RequestBody Pet pet) {

return petRepository.save(pet);

}

// Другие методы (обновление, удаление и т. д.)

}