Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное   
образовательное учреждение высшего образования   
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

Институт информатики и кибернетики

Кафедра технической кибернетики

**Краткая документация по проекту   
«FluffyTailClinic»**

Выполнили:

Рыбалко Ф. А. 6305-010302D

Мурзина А. Е. 6305-010302D

Самара, 2025

## **Лабораторная работа 2: Разработка API**

### **Задачи**

1. Разработка RESTful API
2. Тестирование API с помощью Postman или cURL

**Разработка RESTful API**

**Определение API**

В ходе работы были разработаны REST API для взаимодействия с системой ветеринарной клиники FluffyTailClinic. API включает следующие сущности:

* **User (Пользователь)** — управление пользователями системы (ветеринары, владельцы животных).
* **Animal (Животное)** — информация о питомцах, привязанных к пользователям.
* **Appointment (Запись на приём)** — хранение данных о записях к ветеринарам.
* **Diagnosis (Диагноз)** — справочник возможных диагнозов.
* **DiagnosisAppointment (Связь диагнозов и записей)** — привязка диагнозов к записям на приём.

**Структура API**

**1. API Пользователей (/api/users):**

* GET /api/users — получение списка пользователей
* GET /api/users/{id} — получение информации о пользователе
* POST /api/users/registration — регистрация пользователя
* PUT /api/users/{id} — обновление данных пользователя
* DELETE /api/users/{id} — удаление пользователя

**2. API Животных (/api/animals):**

* GET /api/animals — получение списка животных
* GET /api/animals/{id} — получение информации о конкретном животном
* POST /api/animals — добавление нового животного
* PUT /api/animals/{id} — обновление информации о животном
* DELETE /api/animals/{id} — удаление животного

**3. API Записей на приём (/api/appointments):**

* GET /api/appointments — получение списка записей
* GET /api/appointments/{id} — получение информации о конкретной записи
* POST /api/appointments — создание новой записи
* PUT /api/appointments/{id} — обновление информации о записи
* DELETE /api/appointments/{id} — удаление записи

**4. API Диагнозов (/api/diagnoses):**

* GET /api/diagnoses — получение списка диагнозов
* GET /api/diagnoses/{id} — получение информации о диагнозе
* POST /api/diagnoses — добавление нового диагноза
* PUT /api/diagnoses/{id} — обновление диагноза
* DELETE /api/diagnoses/{id} — удаление диагноза

**5. API Связи диагнозов с приёмами (/api/diagnosis-appointments):**

* GET /api/diagnosis-appointments — получение всех связей диагнозов и записей
* GET /api/diagnosis-appointments/{id} — получение информации о связи
* POST /api/diagnosis-appointments — добавление связи диагноза к записи
* PUT /api/diagnosis-appointments/{id} — обновление связи
* DELETE /api/diagnosis-appointments/{id} — удаление связи

**Разработка DTO для предотвращения зацикливания**

Для предотвращения бесконечной рекурсии в JSON-ответах были использованы **Data Transfer Objects (DTO)**:

* UserDTO — содержит только основные поля пользователя, без списка животных.
* AnimalDTO — содержит ownerId вместо объекта User.
* DiagnosisAppointmentDTO — вместо полных объектов Diagnosis и Appointment содержит их id.

Пример DTO для DiagnosisAppointment:

public class DiagnosisAppointmentDTO {

private Long id;

private Long appointmentId;

private Long diagnosisId;

public DiagnosisAppointmentDTO(DiagnosisAppointment diagnosisAppointment) {

this.id = diagnosisAppointment.getId();

this.appointmentId = diagnosisAppointment.getAppointment().getId();

this.diagnosisId = diagnosisAppointment.getDiagnosis().getId();

}

}

Использование DTO позволило разорвать циклы сериализации и избежать ошибок в API.

**Тестирование API с помощью Postman**

Для тестирования API использовались **Postman**. Были проверены следующие сценарии:

1. **Создание нового пользователя** (POST /api/users/registration)
2. **Добавление животного пользователю** (POST /api/animals)
3. **Запись животного на приём** (POST /api/appointments)
4. **Добавление диагноза к приёму** (POST /api/diagnosis-appointments)
5. **Обновление диагноза** (PUT /api/diagnoses/{id})
6. **Удаление записи на приём** (DELETE /api/appointments/{id})

**Пример запроса в Postman**

POST /api/appointments

{

"animalId": 5,

"vetId": 2,

"date": "2025-03-15T10:30:00",

"description": "Вакцинация"

}

**Пример ответа**

{

"id": 12,

"animalId": 5,

"vetId": 2,

"date": "2025-03-15T10:30:00",

"description": "Вакцинация"

}

**Вывод**

В ходе лабораторной работы было разработано RESTful API для системы управления ветеринарной клиникой FluffyTailClinic. API реализует все основные CRUD-операции, а также использует DTO для предотвращения зацикливания при сериализации. API протестировано с использованием Postman, все запросы работают корректно.