

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**
(БГТУ им. В.Г. Шухова)



ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ

Лабораторная работа №3

по дисциплине: Тестирование программных систем

тема: «Тестирование REST API»

Выполнил: ст. группы ПВ-223

Игнатъев Артур Олегович

Проверили:

Бабенко Анастасия Александровна

Белгород 2025 г.

Лабораторная работа №3

Тестирование REST API

Цель работы: научиться проектировать тестовые сценарии для тестирования REST API.

Вариант 3

Тест-кейсы

1. Добавление питомца (POST /pet).
 - 1.1. Позитивный сценарий (корректные данные).
 - 1.2. Добавление лишних полей.
 - 1.3. Некорректные типы данных.
 - 1.4. Отсутствие обязательных полей.
2. Поиск питомцев по статусу (GET /pet/findByStatus).
 - 2.1. Корректный запрос с одним статусом.
 - 2.2. Корректный запрос с несколькими статусами.
 - 2.3. Некорректный статус.
 - 2.4. Отсутствие параметра status.
 - 2.5. Неизвестный, но валидный формат статуса.

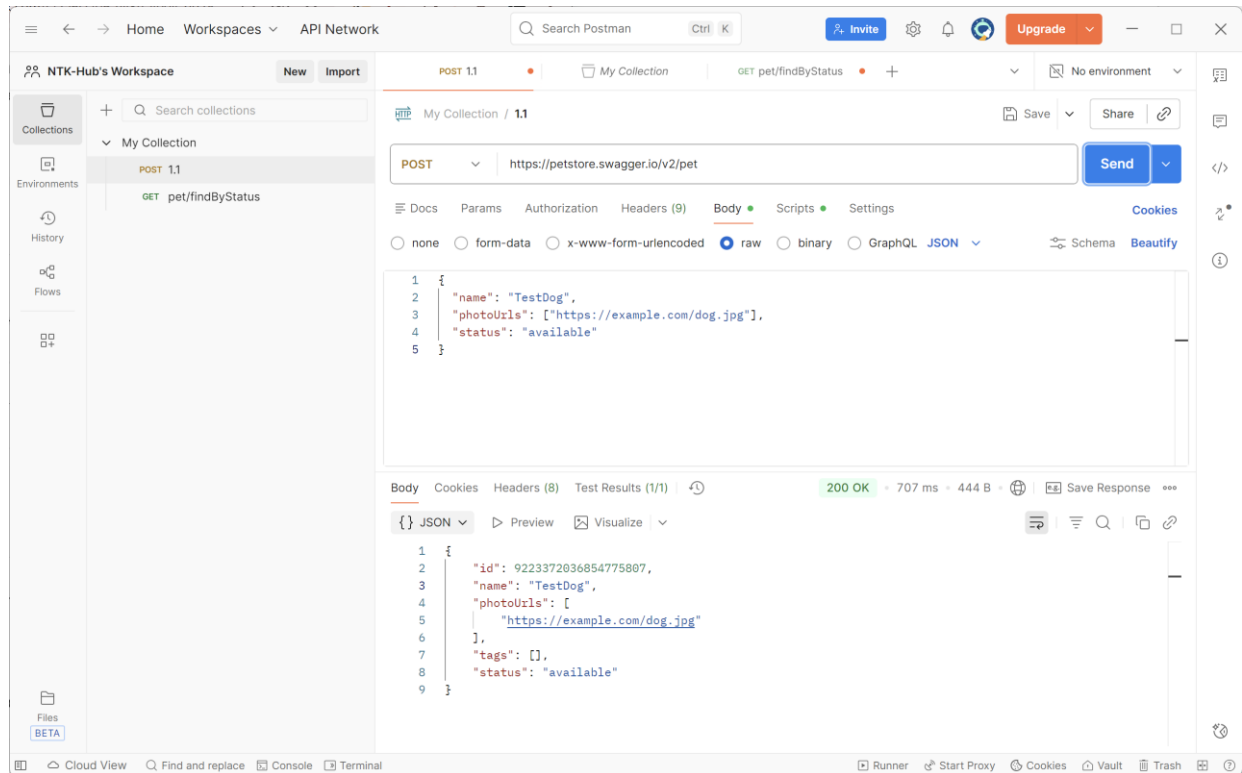
Ход выполнения работы

1. Добавление питомца (POST /pet).

1.1. Позитивный сценарий (корректные данные).

Ожидаемый результат: успешное добавление, код 201.

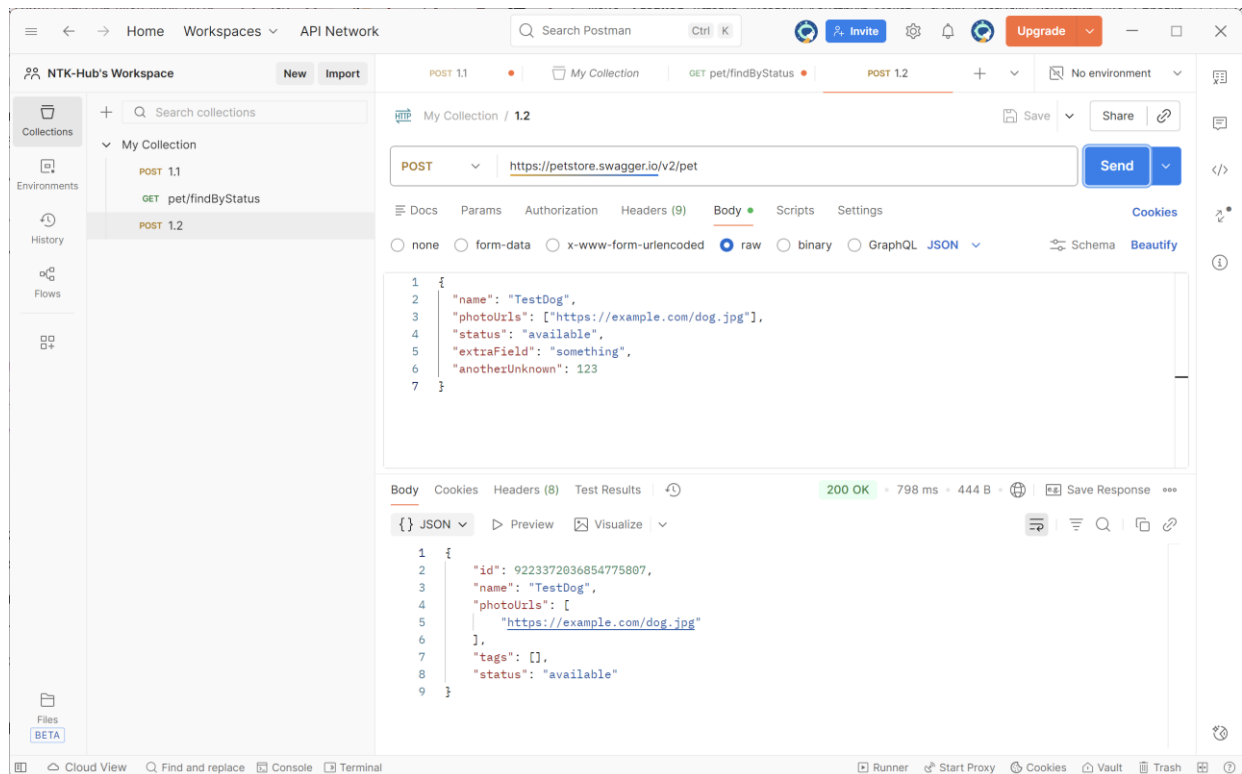
Фактический результат: код 200, питомец добавлен.



1.2. Добавление лишних полей.

Ожидаемый результат: сервер должен возвращать ошибку, если запрос содержит избыточные данные 422.

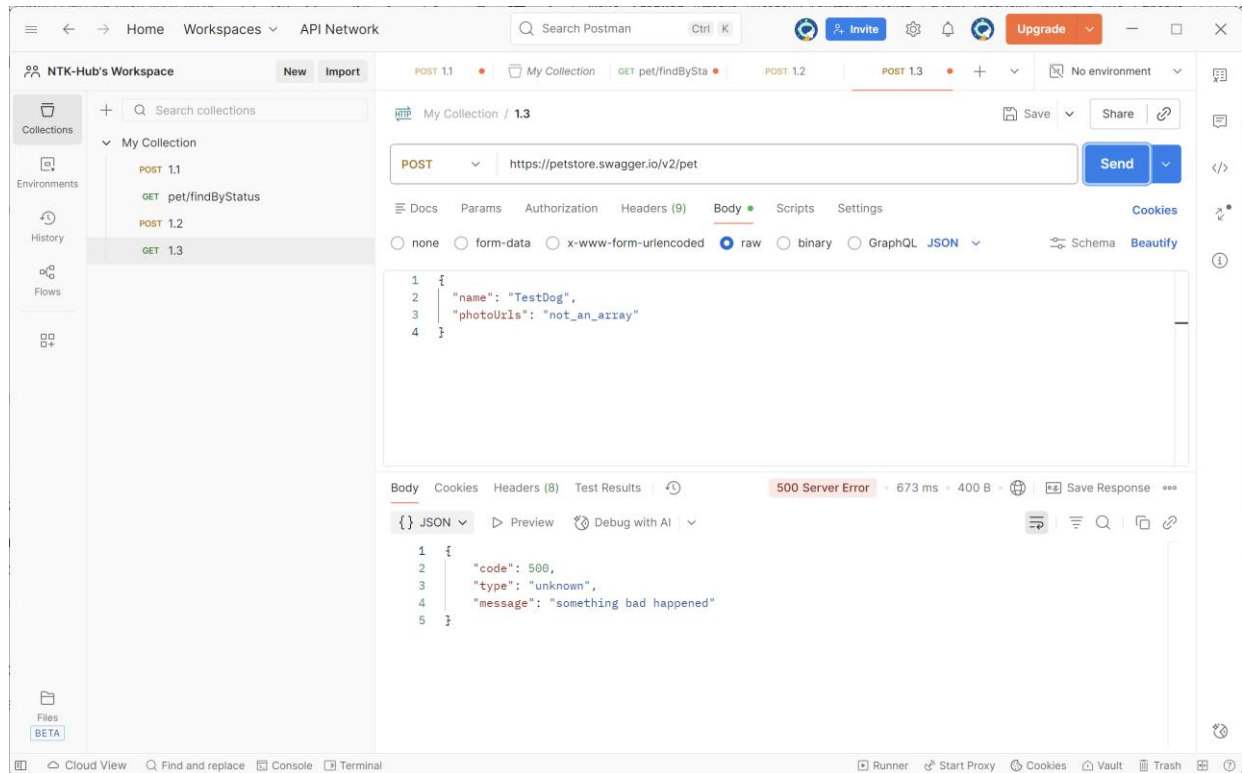
Фактический результат: код 200 ОК, питомец добавлен, лишние поля игнорируются.



1.3. Некорректные типы данных.

Ожидаемый результат: в случае некорректного типа сервер должен ошибку 422.

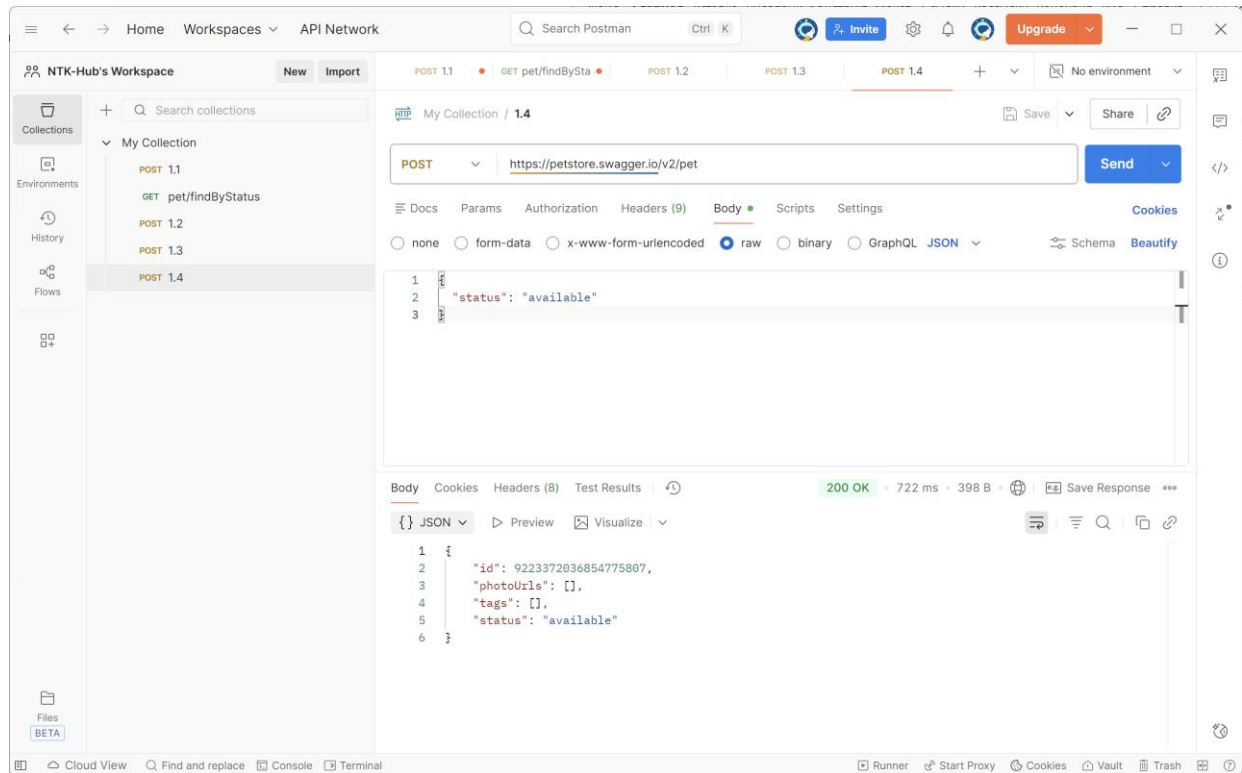
Фактический результат: сервер выдаёт необрабатываемую ошибку с кодом 500.



1.4. Отсутствие обязательных полей.

Ожидаемый результат: при их отсутствии сервер должен выдавать ошибку 422.

Фактический результат: сервер успешно позволяет создать объект без полей вообще.

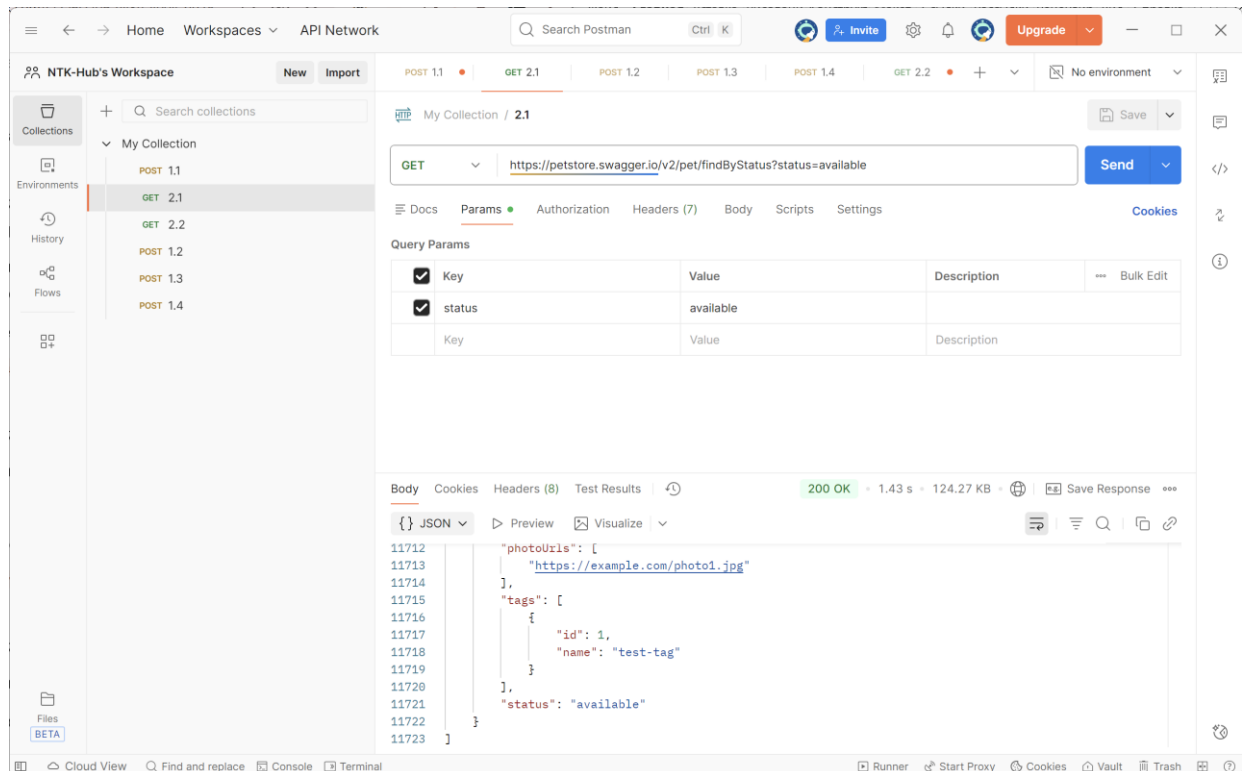


2. Поиск питомцев по статусу (GET /pet/findByStatus).

2.1. Корректный запрос с одним статусом.

Ожидаемый результат: код 200, возвращён большой массив питомцев с указанным статусом.

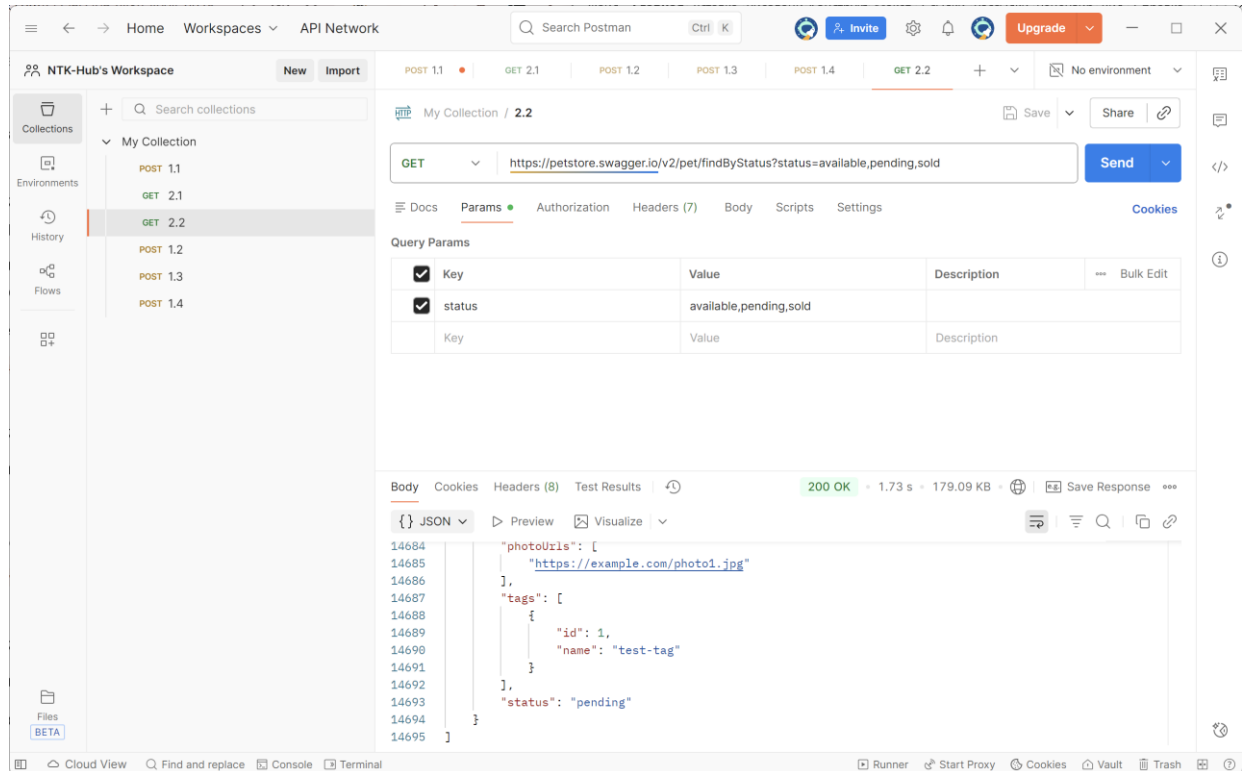
Фактический результат: код 200, возвращён массив из тысяч объектов.



2.2. Корректный запрос с несколькими статусами.

Ожидаемый результат: код 200, массив содержит питомцев всех указанных статусов.

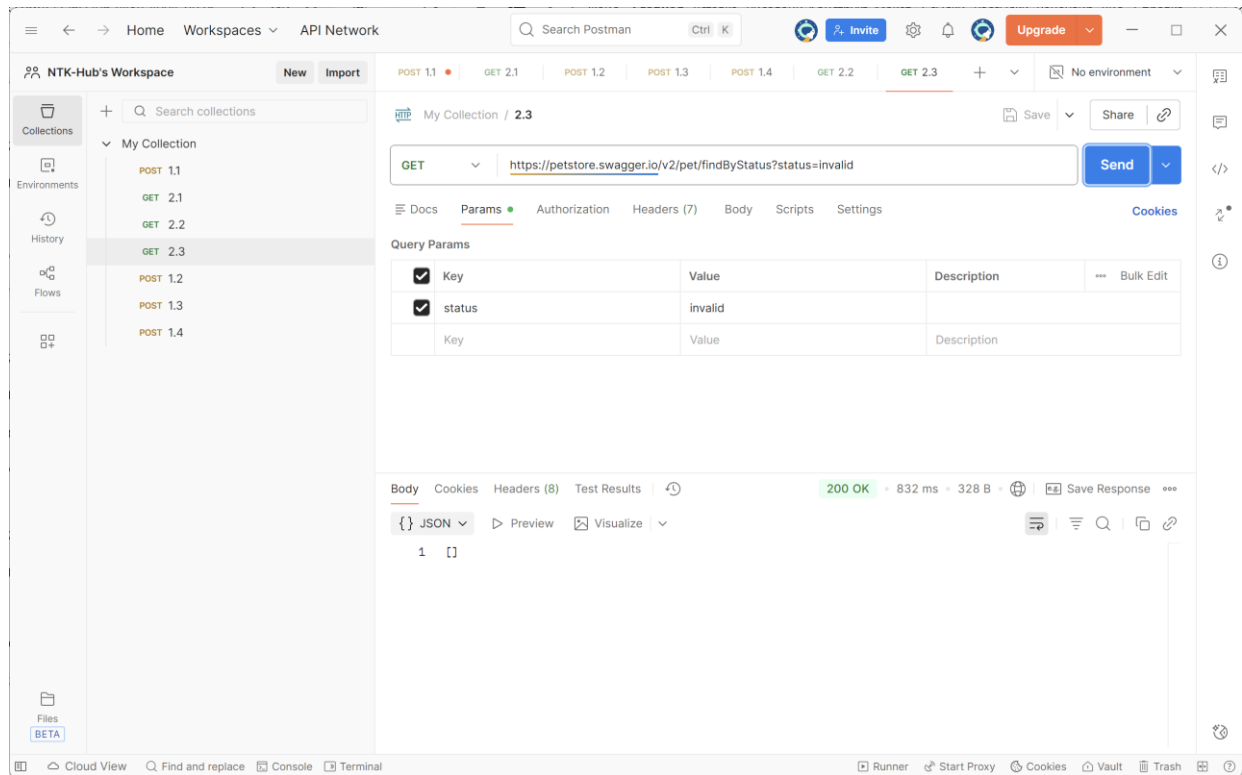
Фактический результат: код 200, возвращён очень большой массив.



2.3 Некорректный статус.

Ожидаемый результат: сервер должен выдавать ошибку 400.

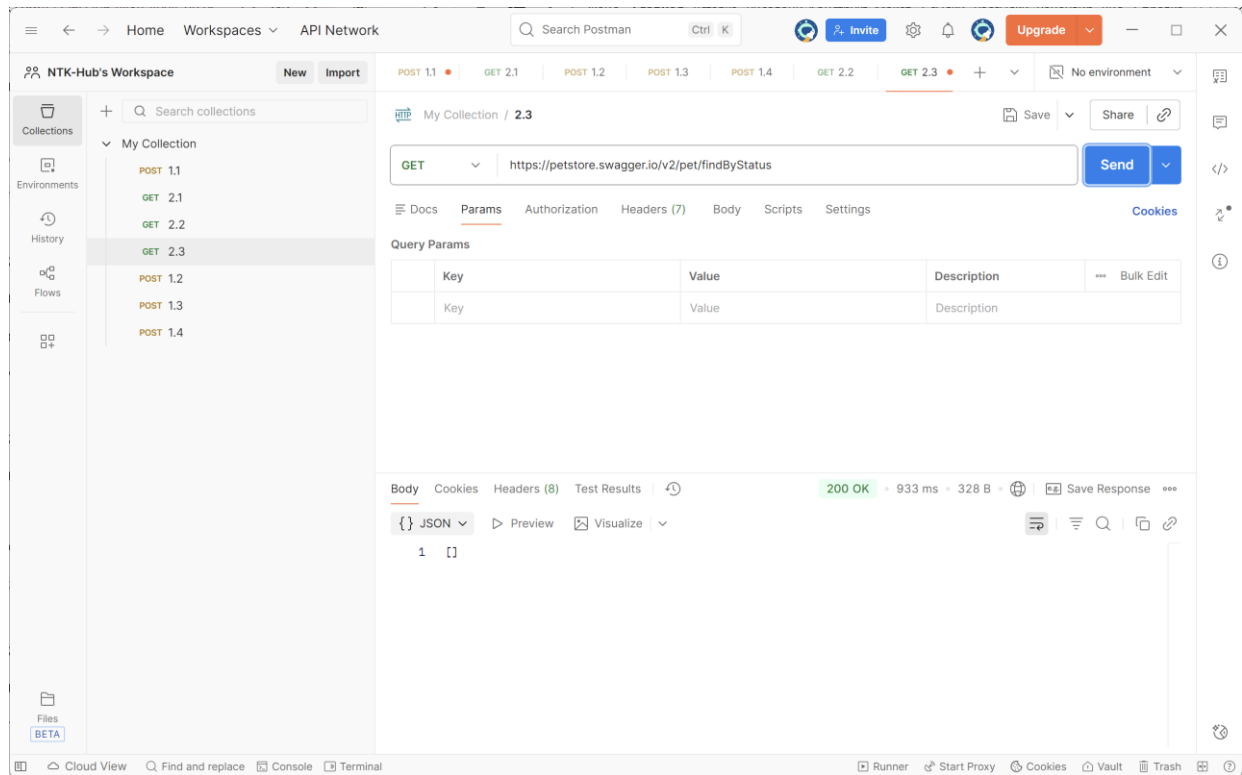
Фактический результат: код 200, возвращён пустой массив [].



2.4. Отсутствие параметра status.

Ожидаемый результат: сервер должен выдавать ошибку 400.

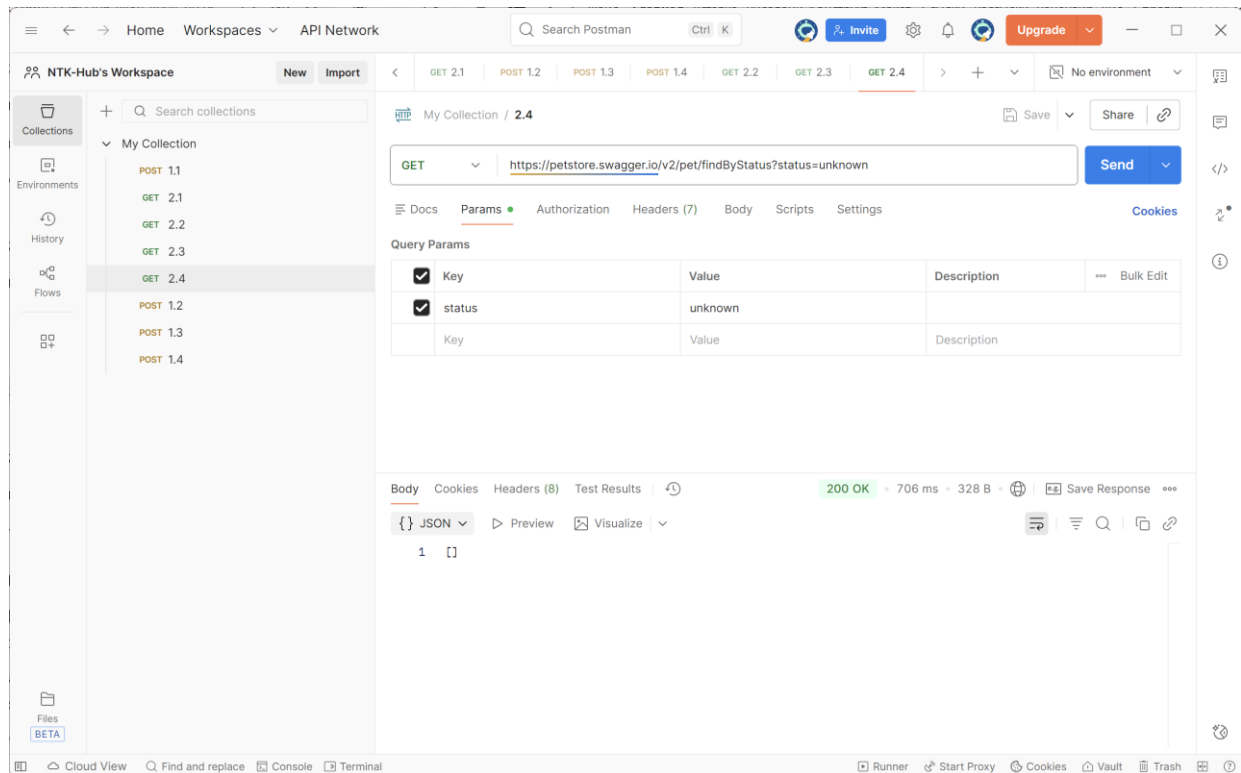
Фактический результат: код 200, возвращён пустой массив [].



2.5. Неизвестный, но валидный формат статуса.

Ожидаемый результат: код 200, пустой массив [].

Фактический результат: код 200, возвращён пустой массив [].



Вывод: в ходе лабораторной работы научились проектировать тестовые сценарии для тестирования REST API. Познакомились с инструментом тестирования REST API - Postman.