**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая кибернетика и информационные технологии»

**Отчет по практической работе**

по дисциплине «Основы работы с протоколом HTTP»

Выполнила: студентка группы БПИ2401 Рябова Екатерина

Проверил: Мкртчян Грач Маратович

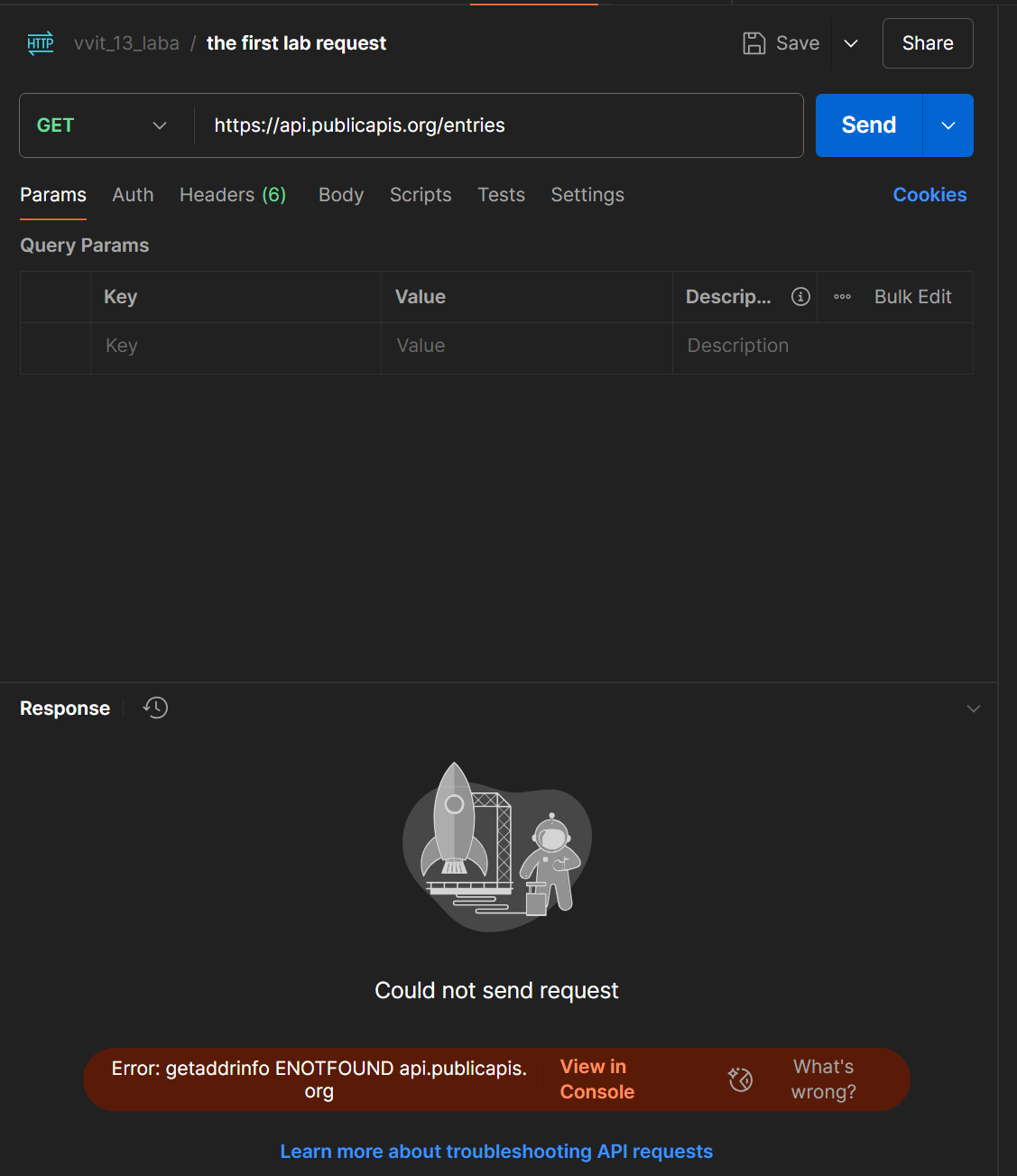
Москва

2025

1. **Цель работы:**

Получить практические навыки в использовании Postman для тестирования и разработки API, изучить основные методы HTTP, научиться создавать и отправлять запросы, а также анализировать ответы сервера. По завершении каждого задания студенты должны сделать скриншоты своих запросов и ответов для включения в отчет.

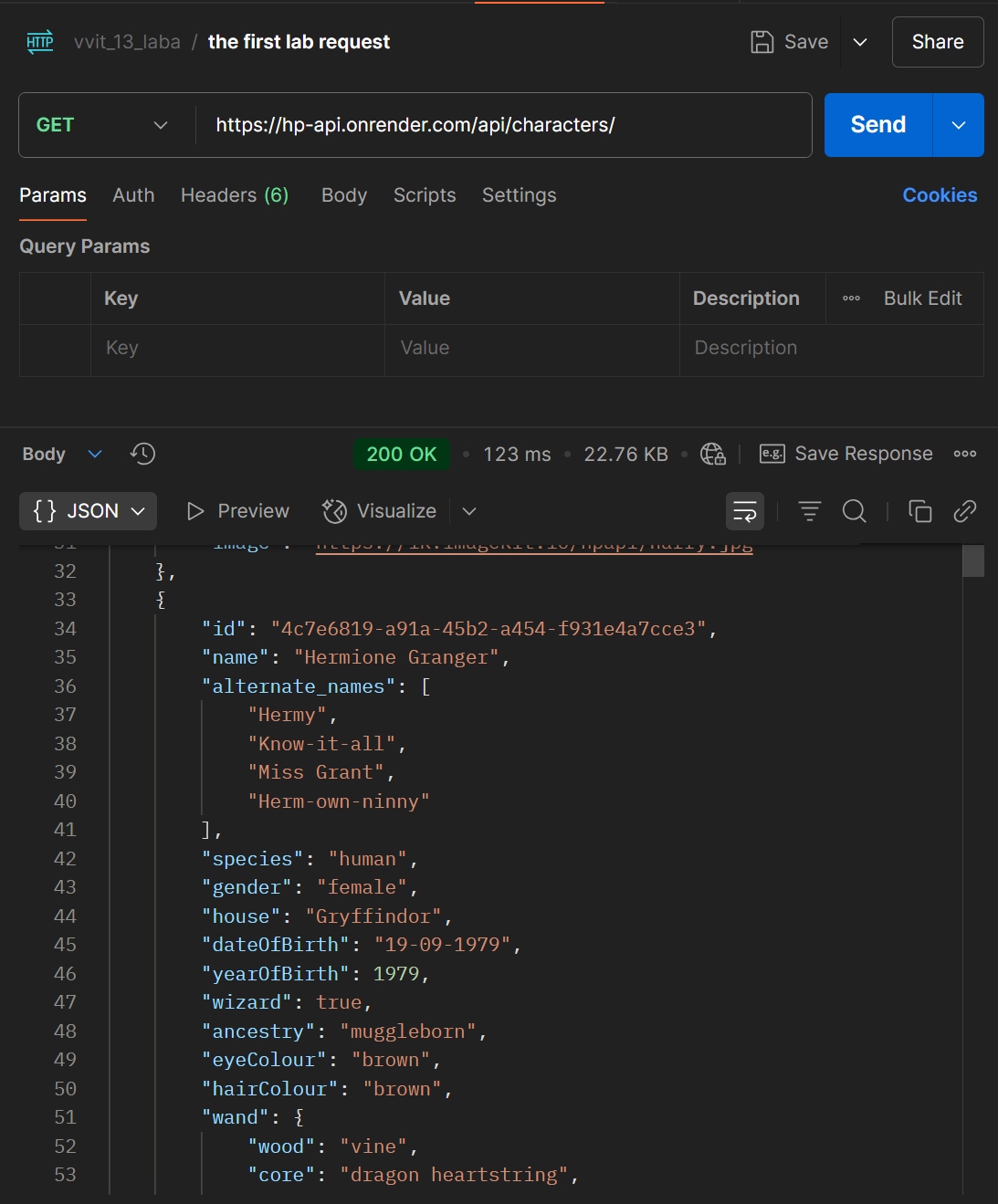
1. **Ход работы:**
2. Установим Postman.
3. Создадим новый запрос и сохраним его в коллекцию «vvit\_13\_laba».
4. Выберем метод GET, введем предложенные адрес API и убедимся, что нам выдается ошибка:



Предположим род ошибки:

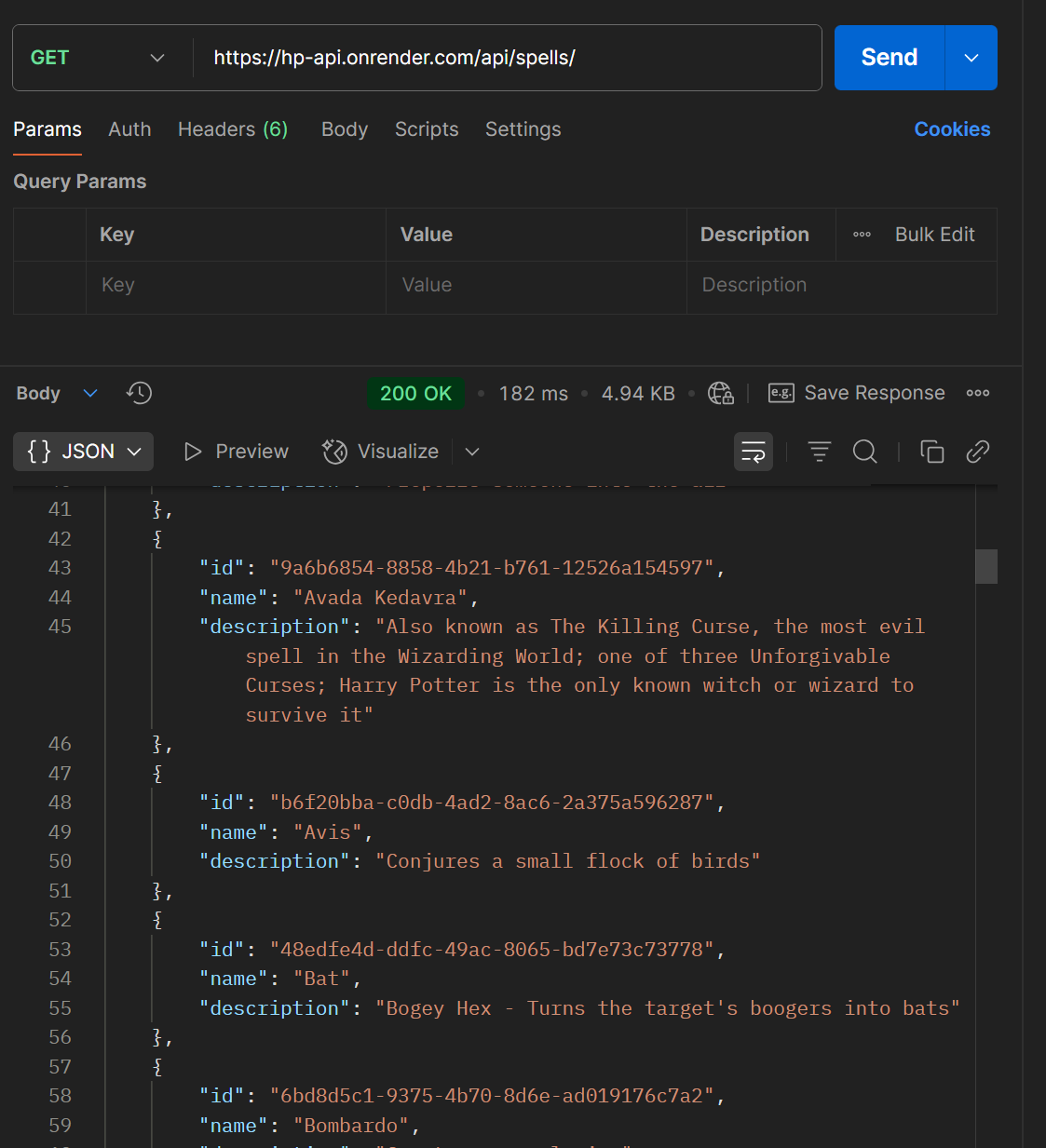
* Неправильный URL-адрес
* Недоступность сервера.

1. Поищем другие доступные API, например, по Гарри Поттеру. Я нашла такую: https://hp-api.onrender.com/

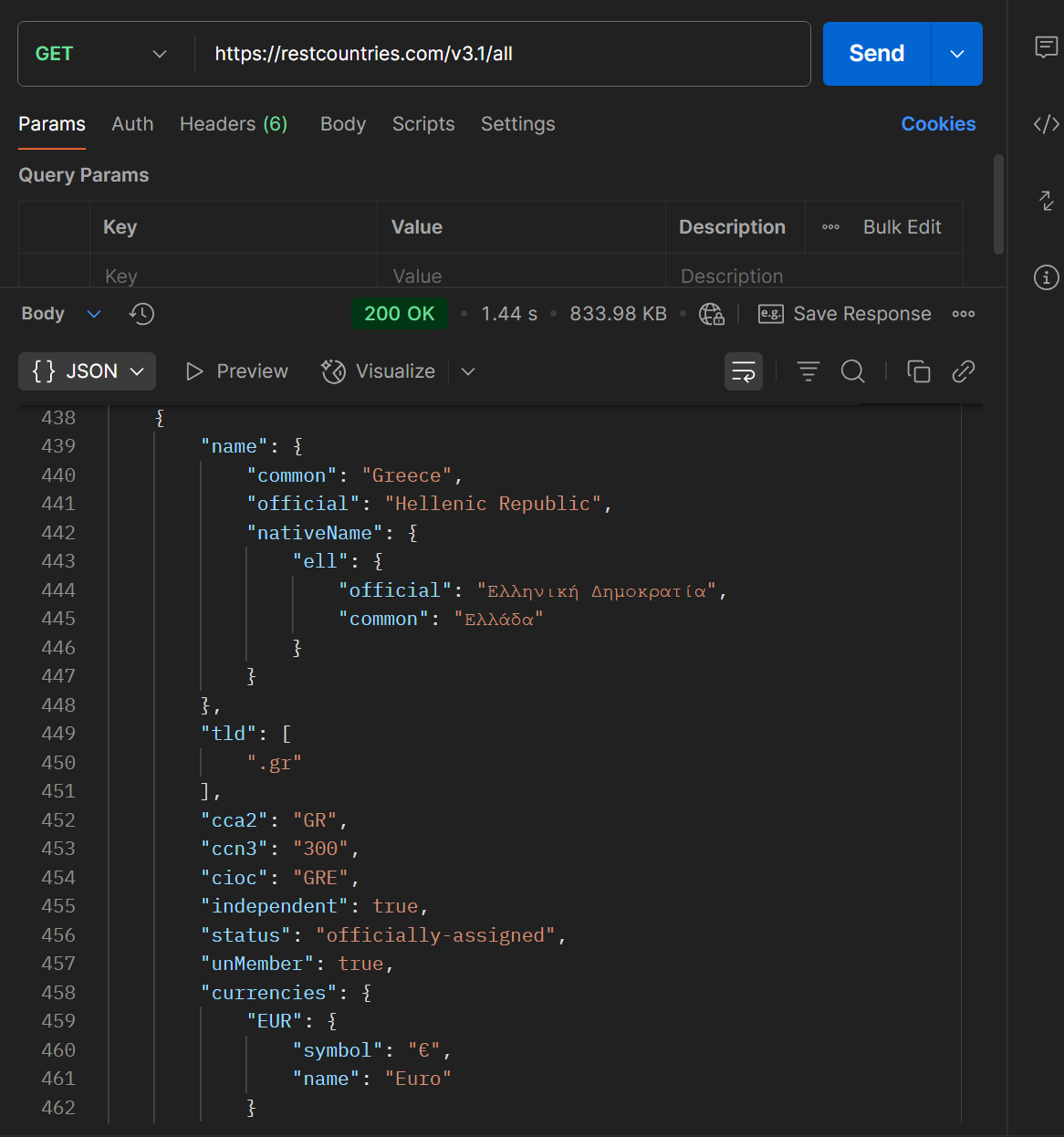


Выберем JSON и посмотрим, какой нам приходит ответ.

Попробуем так же несколько разных запросов – персонажей и заклинаний. Статус-код ответа – 200ОК

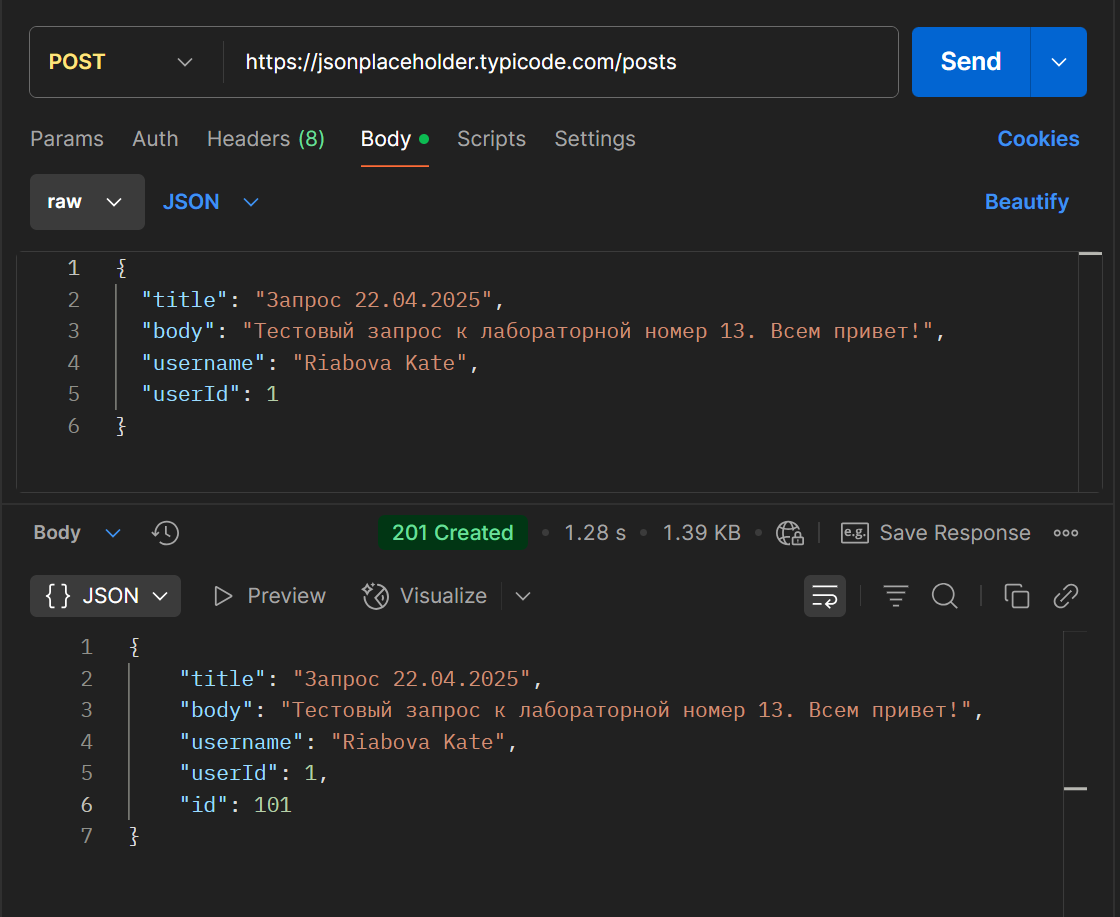


1. Теперь выберем REST Countries API (URL: https://restcountries.com/v3.1/all) для получения информации о всех странах. Создадим новый GET-запрос и посмотрим на результат в формате JSON:



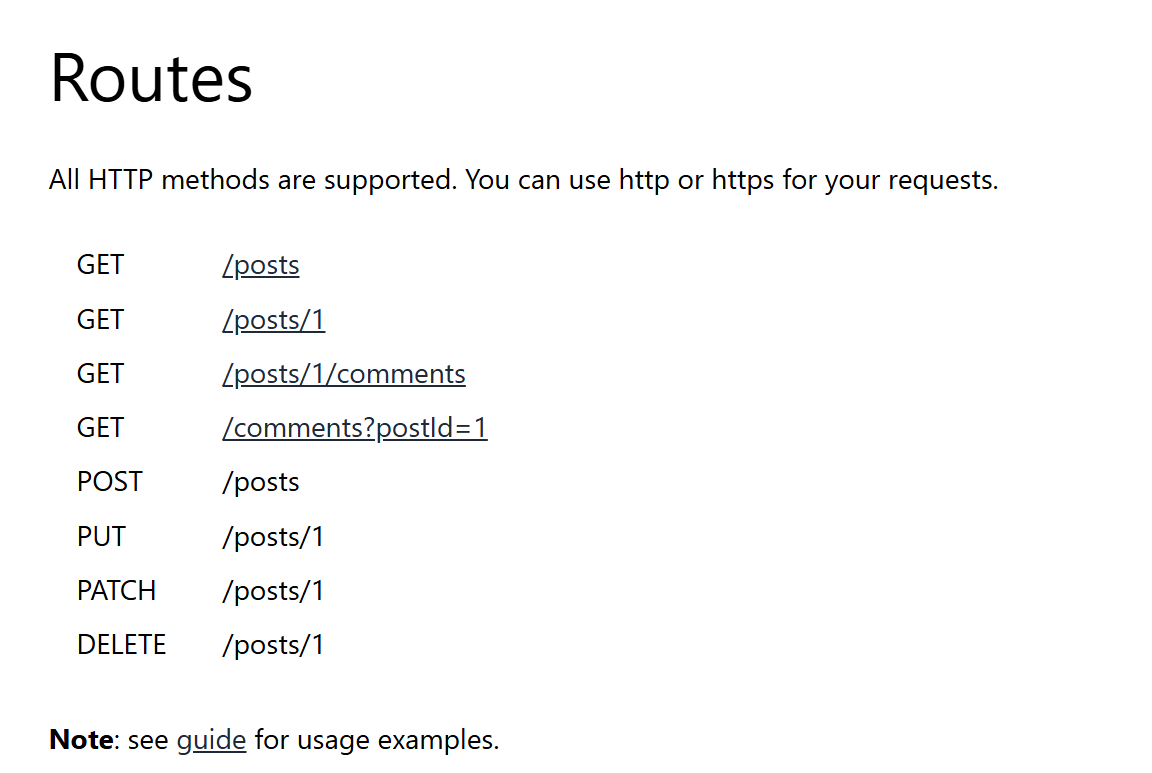
Статус-код ответа – 200 ОК

1. Используем JSONPlaceholder (URL: https://jsonplaceholder.typicode.com/posts) для отправки тестовых запросов.
2. Запрос POST – создание поста:



Статус-код ответа – 201 Created, присвоен id посту - 101

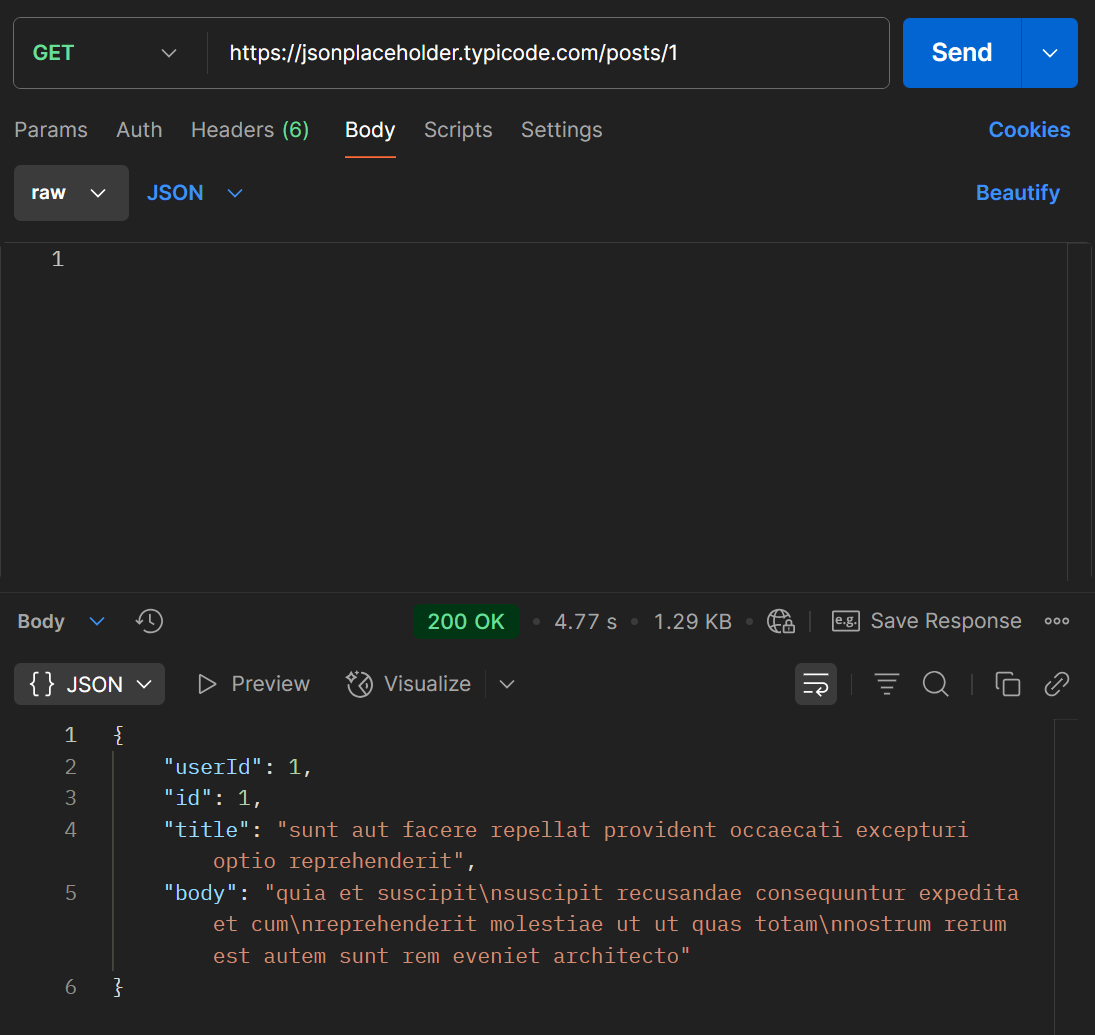
1. Запрос PUT – обновление существующего поста. На сайте <https://jsonplaceholder.typicode.com/> почитаем инструкции:



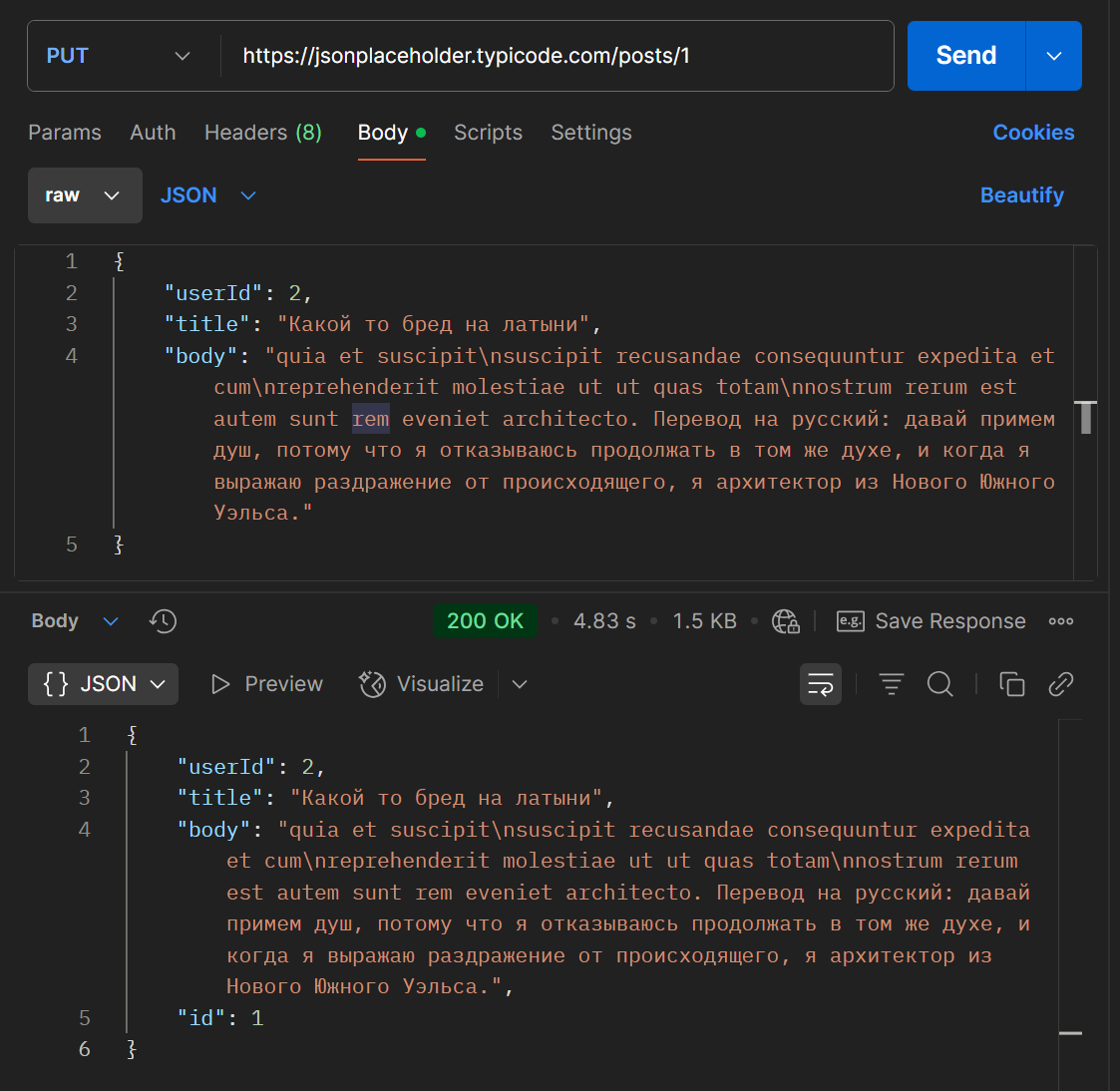
Таким образом, при PUT-запросе нужно использовать /post/1 и передавать все поля, даже если меняете только одно. Иначе отсутствующие поля могут быть перезаписаны на null.

Можно изменить любые поля существующего поста: title, body, userId.

Вот что изначально лежит в post/1:



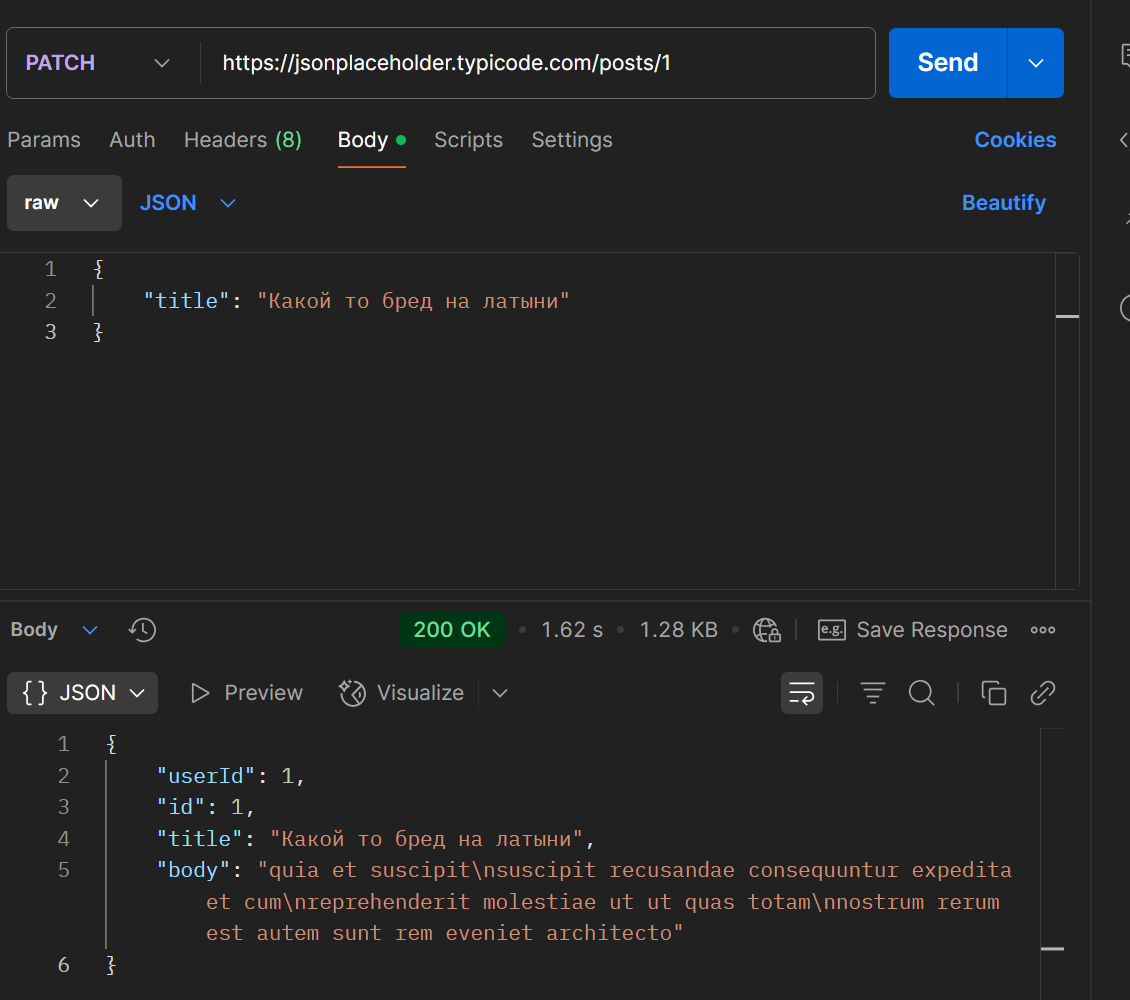
Изменим userId, title и body.



Статус-код ответа – 200ОК, id поста не меняется.

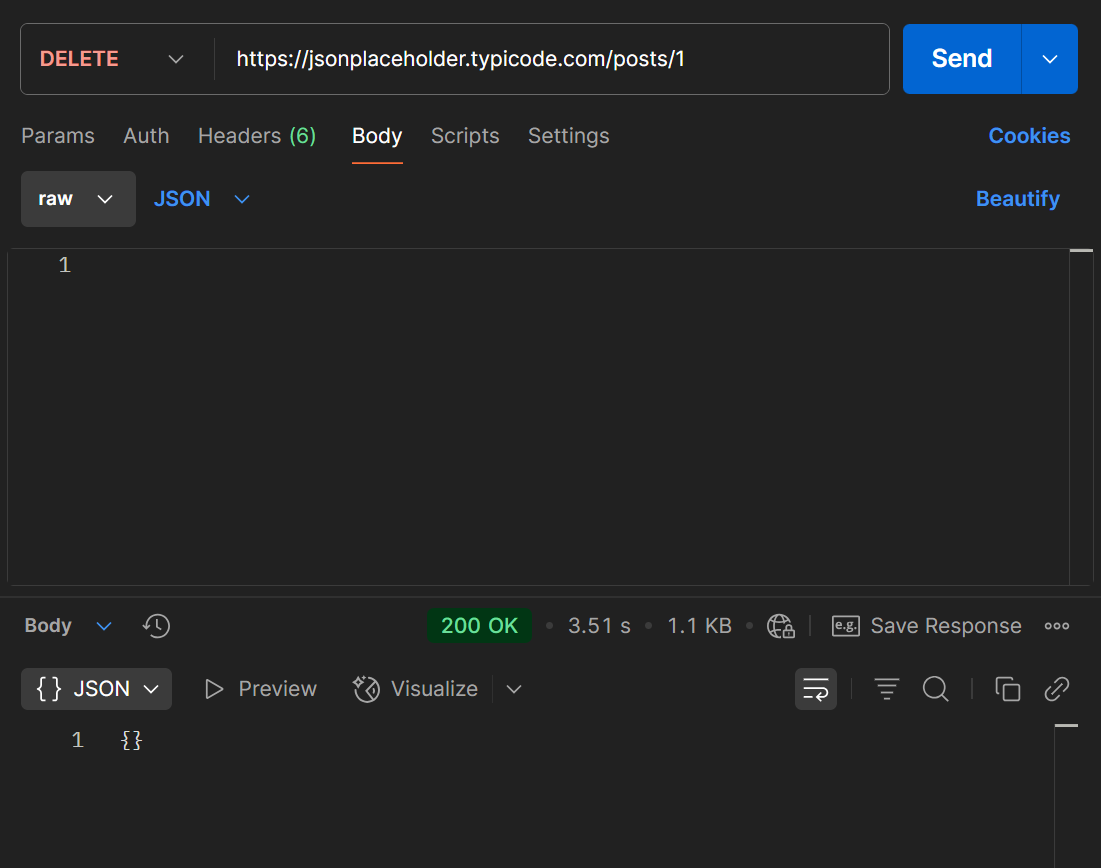
1. Проверим запрос PATCH – требует передачи всех полей (даже неизменяемых).

Если нужно обновить только одно поле (например, title), лучше использовать PATCH



Статус-код сервера – 200.

1. Запрос DELETE – удаление. Убедимся, что в инструкции тоже нужно использовать существующий пост 1.



Статус-код ответа – 200ОК. Ответ – постой JSON

Почему при повторной отправке запроса GET на post/1 у нас выдается то, что было, без изменений? Потому что JSONPlaceholder не сохраняет данные на сервере, но имитирует успешный ответ.

В реальных API для DELETE может потребоваться авторизация (API-Key, токен).

1. **Вывод:**

Успешно получены практические навыки в использовании Postman, мы научились создавать и отправлять запросы, а также анализировать ответы сервера