**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»

Лабораторная работа №14

Основы работы с протоколом HTTP

Выполнила: Студентка группы

БВТ2402

Шилинцева Татьяна

Москва

2025

**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 сделать скриншоты своих запросов и ответов для включения в отчёт. Провести краткий анализ полученных данных и сделать выводы о выполненной работе.

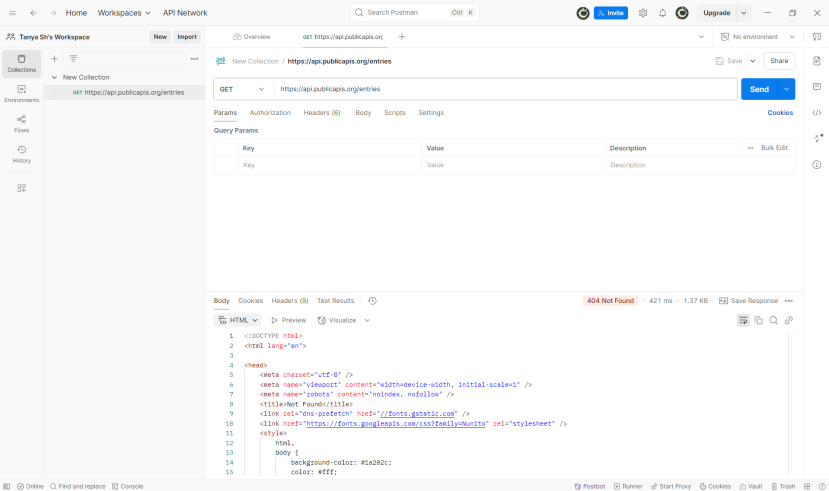
***Задание 1*** Ознакомление с Postman

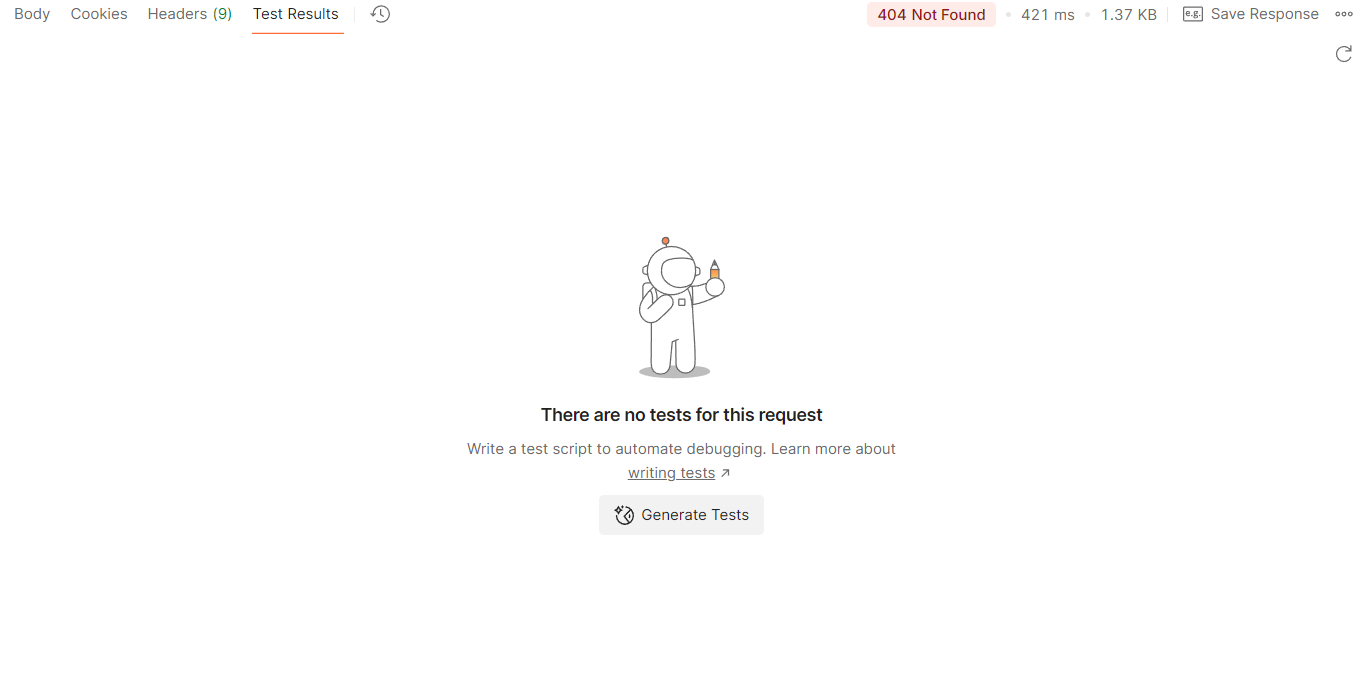
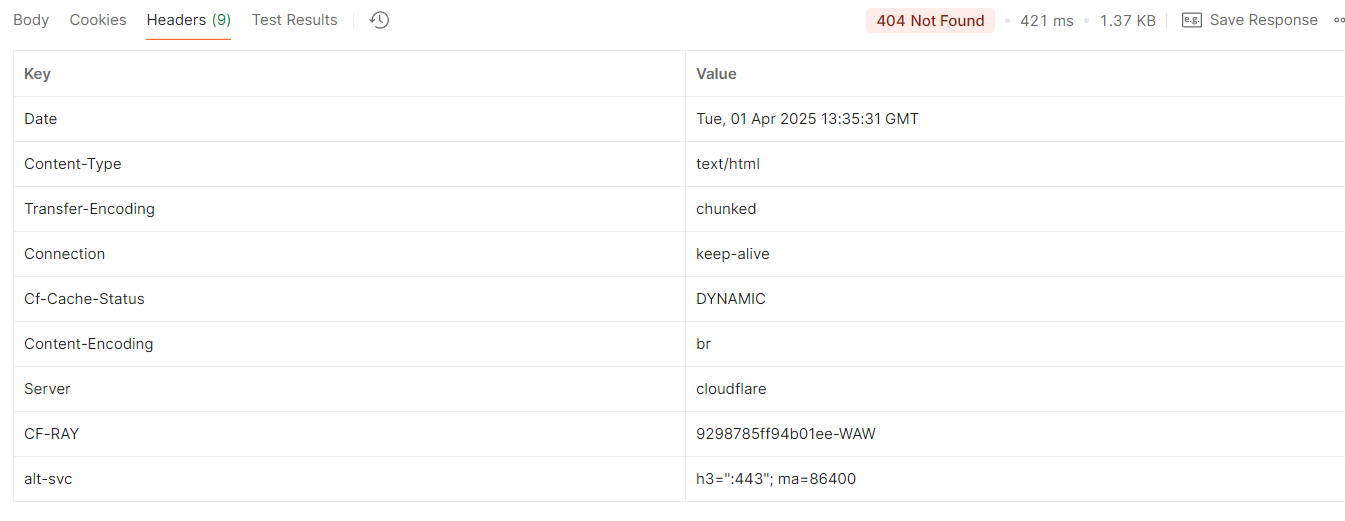
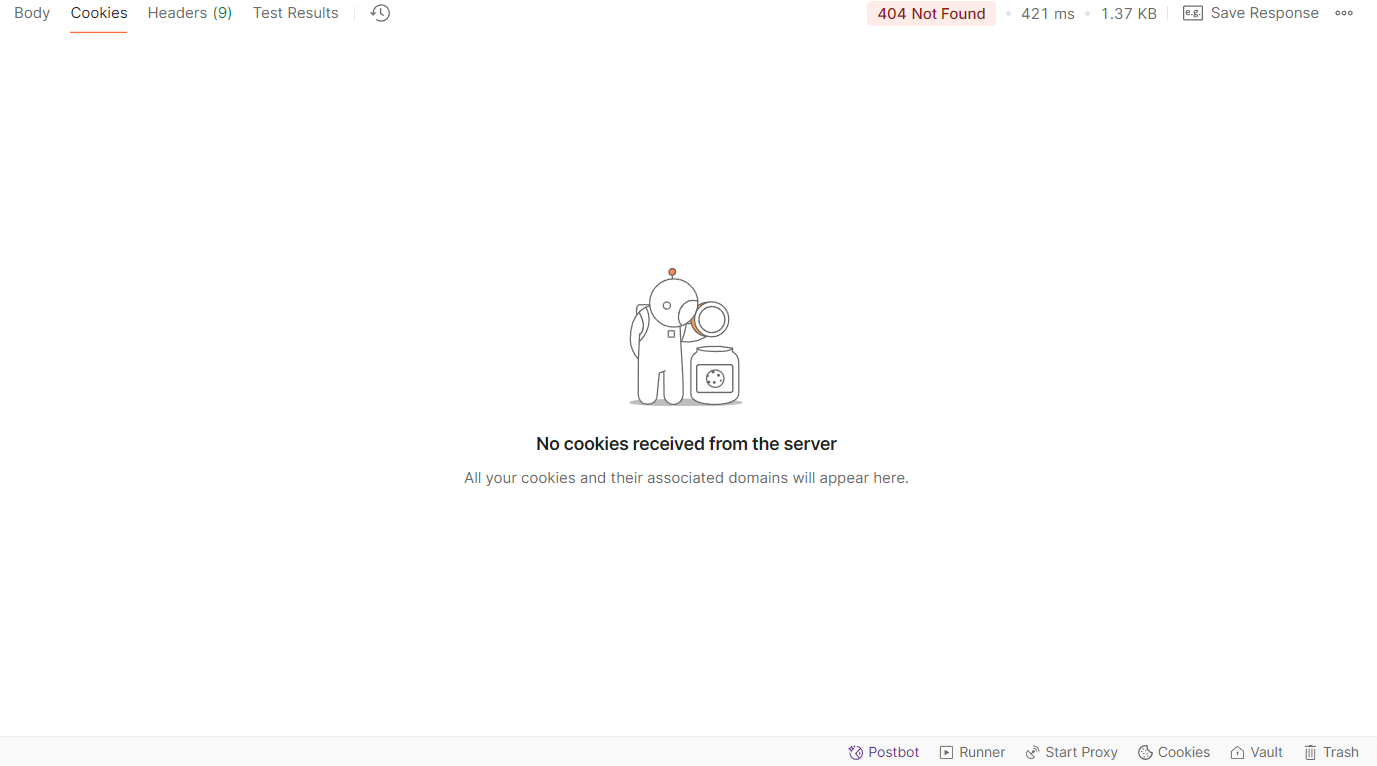
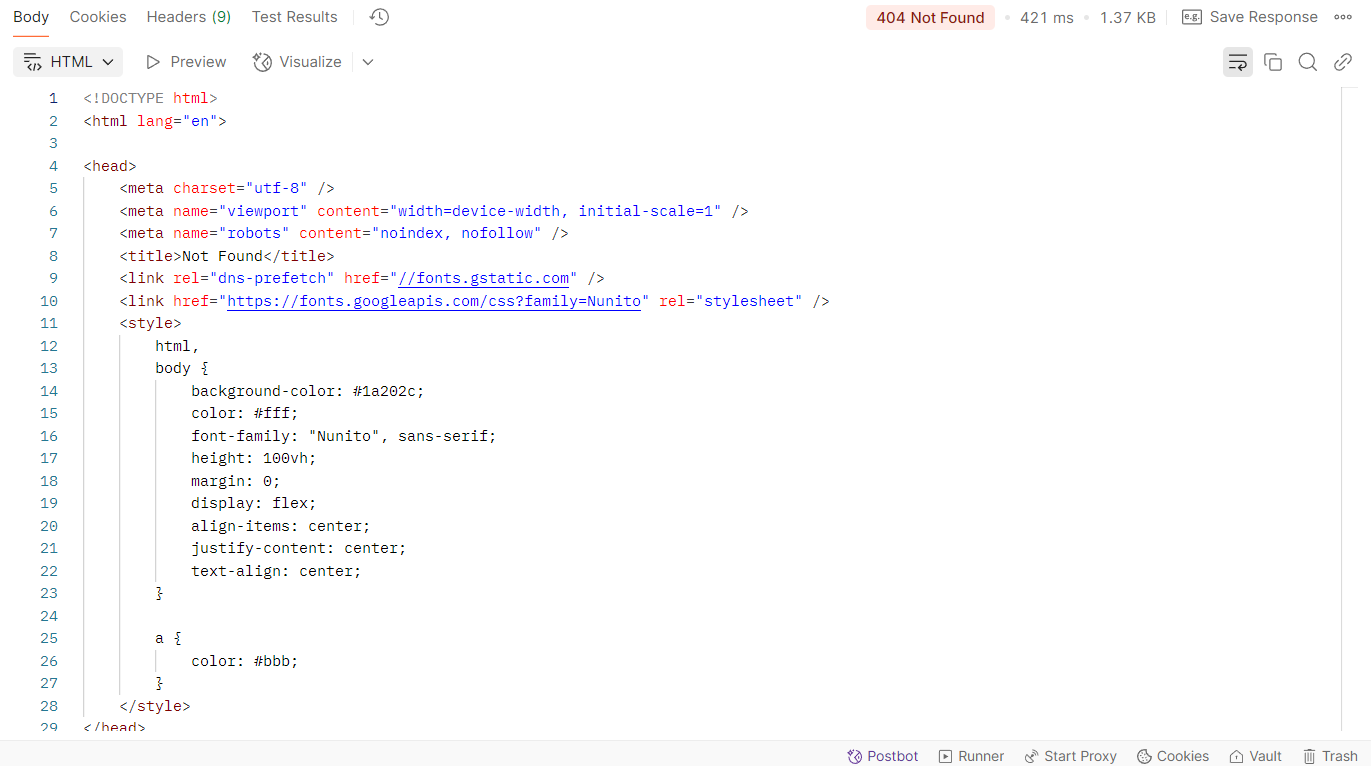
Инструкция:

1. Установка Postman. Перейти на официальный сайт Postman и скачать последнюю версию приложения для вашей операционной системы.
2. Установить и открыть Postman.
3. Создание первого запроса. В интерфейсе Postman нажать на кнопку "**New**" и выбрать "**Request**". Дать запросу название и сохранить его в новую или существующую коллекцию. В поле ввода URL ввести адрес публичного API (например, **https://api.publicapis.org/entries**) и выбрать метод **GET**. Нажать на кнопку "**Send**" для отправки запроса и просмотреть ответ сервера внизу экрана.

Выполнение:

Запрос:



Получено:

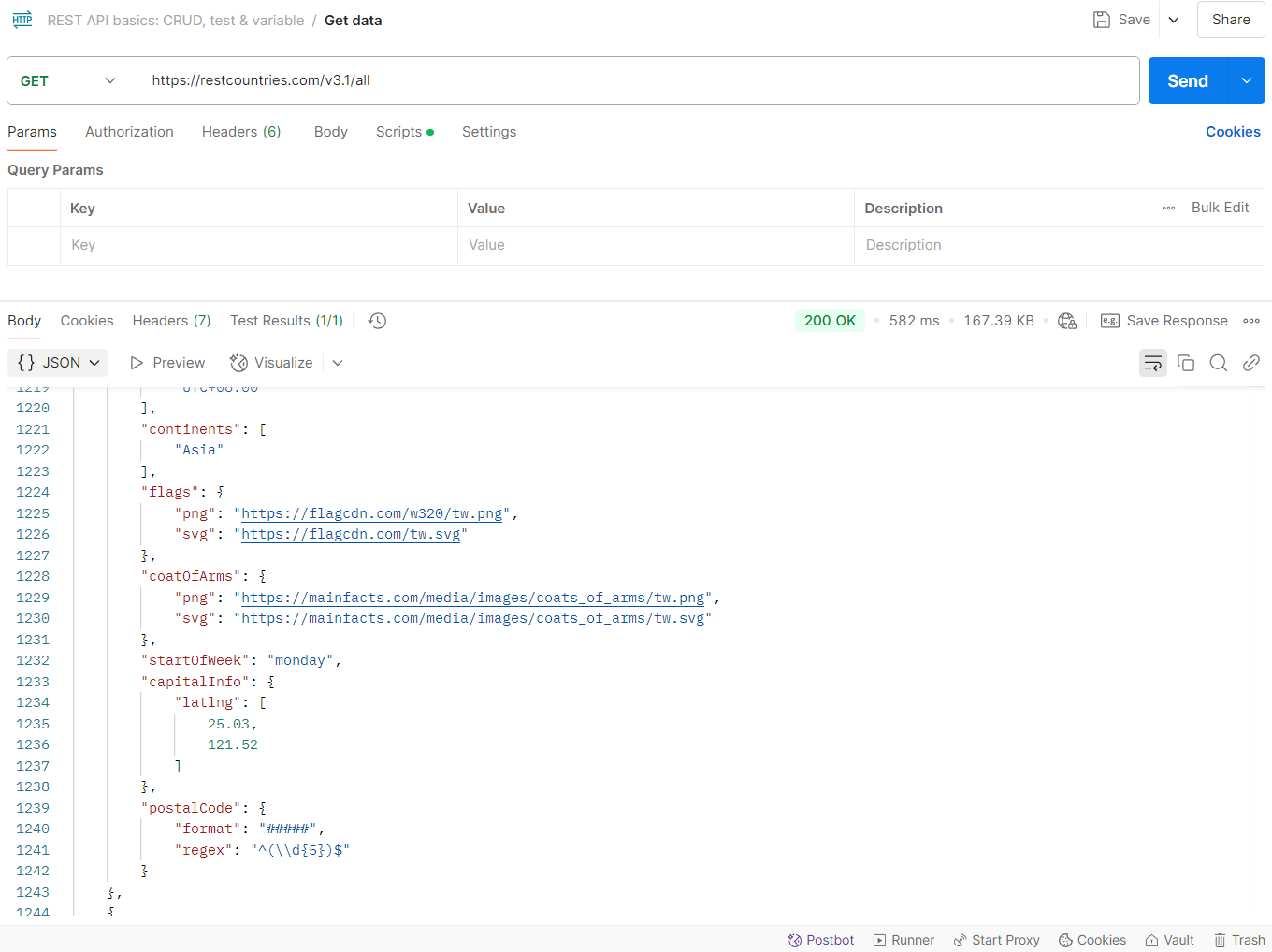
Анализ: В результате отправки запроса в качестве ответа был получен код разметки сайта, на который указывает данная ссылка, информация о файлах Cookies, краткую информацию в табличном формате в разделе Headers, а также Test Results.

***Задание 2*** Работа с публичным API

Инструкция:

1. Выбрать **REST Countries API (URL: https://restcountries.com/v3.1/all)** для получения информации о всех странах.
2. Создать новый GET запрос к выбранному API.
3. Нажать "Send" и изучить структуру ответа в формате JSON.

Выполнение:

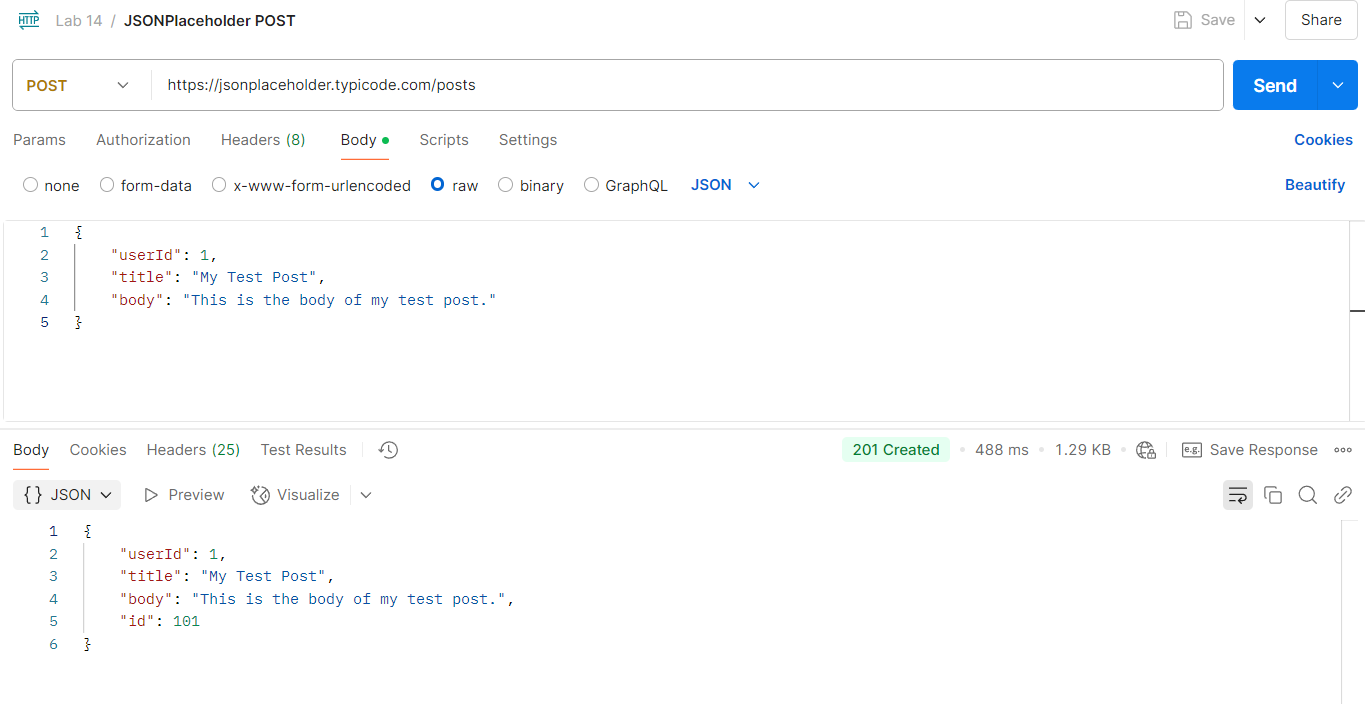


Анализ: В результате данного запроса были получены данные формата JSON. Их структура представляет собой массу вложенных друг в друга словарей и списков. Помимо всего прочего мы можем увидеть время выполнения запроса 582 ms и статус 200 ОК.

***Задание 3*** Отправка данных с использованием POST

Инструкция:

1. Использовать JSONPlaceholder (URL: https://jsonplaceholder.typicode.com/posts) для отправки тестовых POST запросов.
2. Переключиться на метод POST и ввести URL для создания поста.
3. В "Body" выбрать тип "raw" и формат JSON, ввести тело запроса.
4. Отправить запрос и проанализировать ответ сервера.

Выполнение:

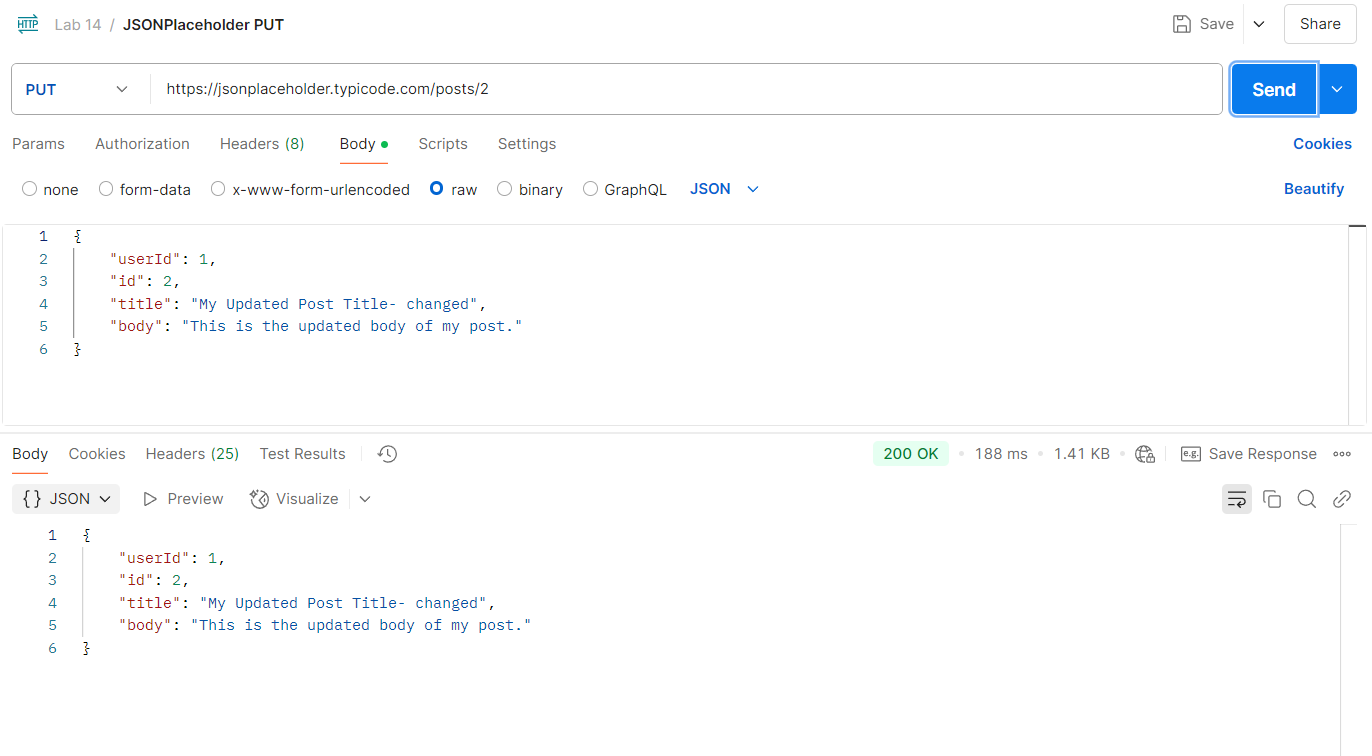
Анализ: После введения запроса был получен ответ сервера, содержащий дополнения к содержимому, указанному в Body. Статус запроса 201 Created (это код состояния HTTP, который указывает на успешное выполнение запроса и создание нового ресурса), время выполнения составило 488 ms.

***Задание 4*** Модификация данных с использованием PUT

Инструкция:

1. Использовать JSONPlaceholder, выбрать существующий пост для обновления.
2. Изменить метод на PUT и указать полный URL для обновления поста.
3. В "Body" изменить содержание поста, отправить запрос и проанализировать ответ.

Выполнение:



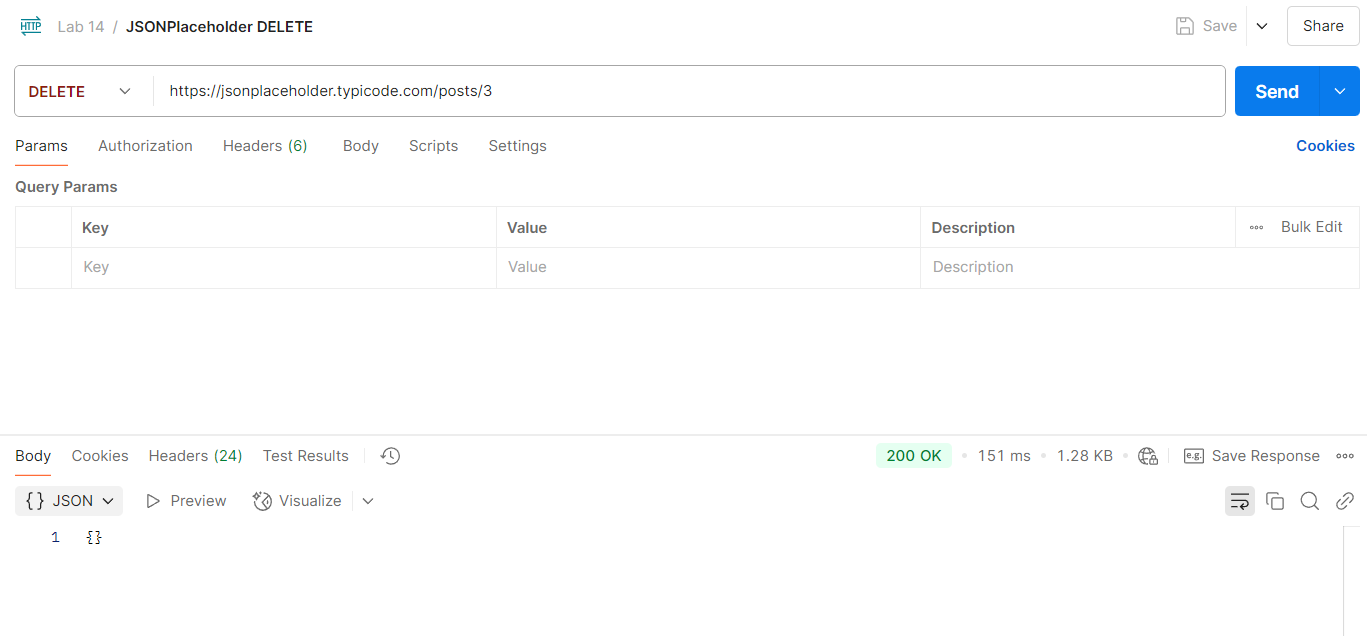
Анализ: После выполнения запроса типа PUT с изменением изначального содержания запроса (в “title” были добавлены символы “- changed”), пост был успешно обновлён, о чём нам говорит статус 200 ОК, время выполнения запроса составило 188 ms.

***Задание 5*** Удаление данных с использованием DELETE

Инструкция:

1. Использовать JSONPlaceholder, выбрать пост для удаления.
2. Изменить метод на DELETE и указать URL поста для удаления.
3. Отправить запрос и проверить статус-код ответа сервера.

Выполнение:



Анализ: В результате выполнения запроса DELETE был получен пустой ответ JSON формата. Статус запроса 200 ОК. Время выполнения запроса составило 151 ms.

**Вывод**: В ходе выполнения данной работы были получены знания о том, как работать с HTTP запросами различных форматов (GET, POST, PUT, DELETE) с помощью программы Postman для тестирования и разработки API.