МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное учреждение высшего образования

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

Лабораторная работа №14

**Основы работы с протоколом HTTP**

**по дисциплине**

**«Введение в информационные технологии»**

Выполнил: студент гр. БВТ2403

Лашков Ю.Е.

Проверил: Павликов А.Е.

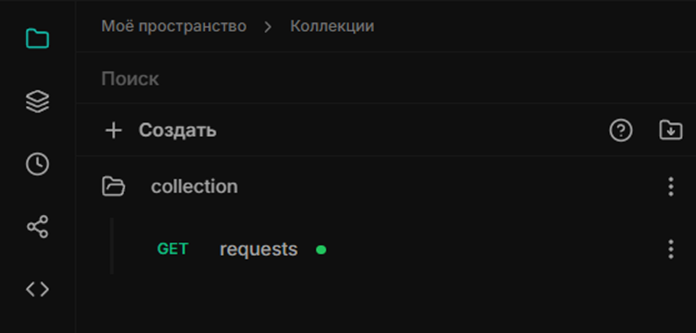
Москва, 2025 г.

**Цель работы:**Закрепить навыки работы с HTTP-протоколом и API-запросами, используя инструмент Postman. Практически освоить основные методы HTTP (GET, POST, PUT, DELETE), научиться создавать запросы, анализировать ответы сервера и работать с публичными API.

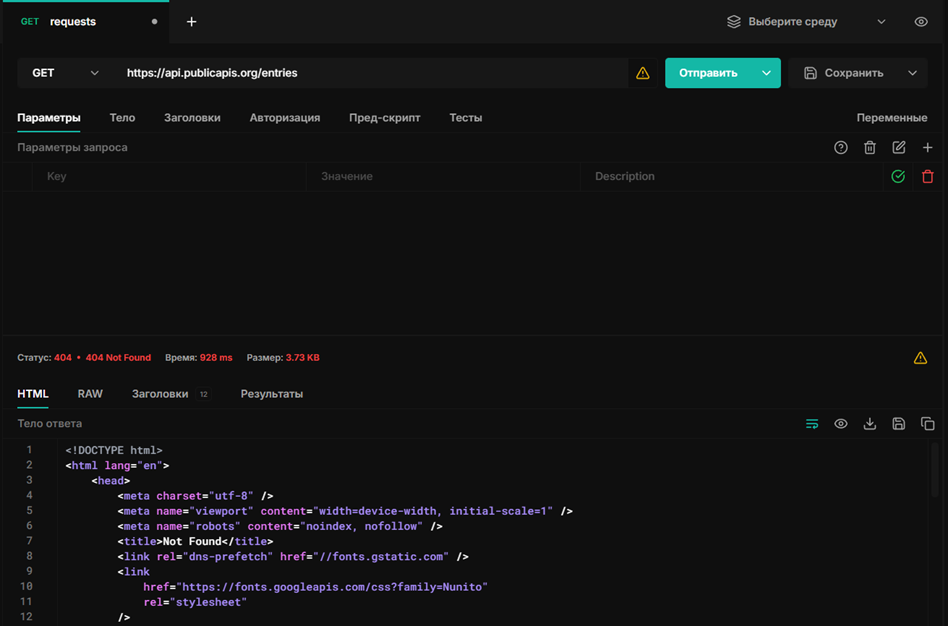
**Оборудование:**1. Компьютер с установленной операционной системой.2. Программа Postman или Hoppscotch для тестирования API.

**Введение:**HTTP-протокол и API являются основой современных веб-технологий, позволяя клиентам и серверам обмениваться данными. С помощью инструментов, таких как Postman или Hoppscotch, можно тестировать запросы (GET, POST, PUT, DELETE) и анализировать ответы сервера. В данной работе мы будем использовать Postman для взаимодействия с публичными API, чтобы освоить создание запросов, изучить структуру JSON-ответов и понять принципы работы RESTful API.

**Практическая работа**

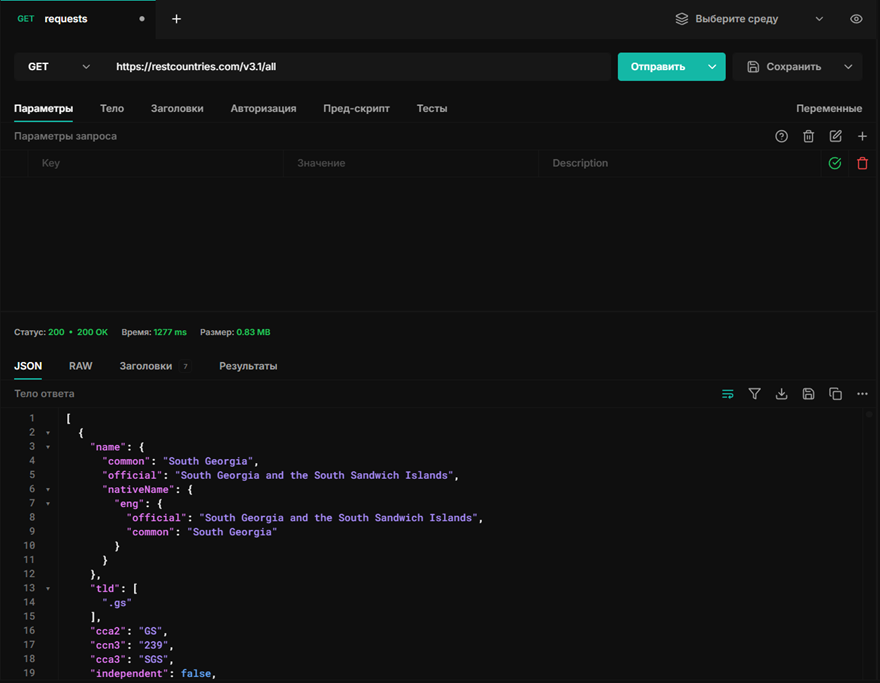
**1. Ознакомление с Postman (Hoppscotch):**

Здесь я создал новую коллекцию в Hoppscotch с названием collection. Затем добавил новый запрос, который назвал requests, и сохранил его в эту коллекцию. Это позволит мне структурировать и организовать запросы для удобства дальнейшего тестирования API.



Здесь я отправил GET-запрос к URL https://api.publicapis.org/entries. После отправки запроса получил ответ с статусом 404 Not Found, что говорит о том, что запрашиваемый ресурс не был найден на сервере. В теле ответа содержится HTML-код, который обычно отображается при ошибке 404.

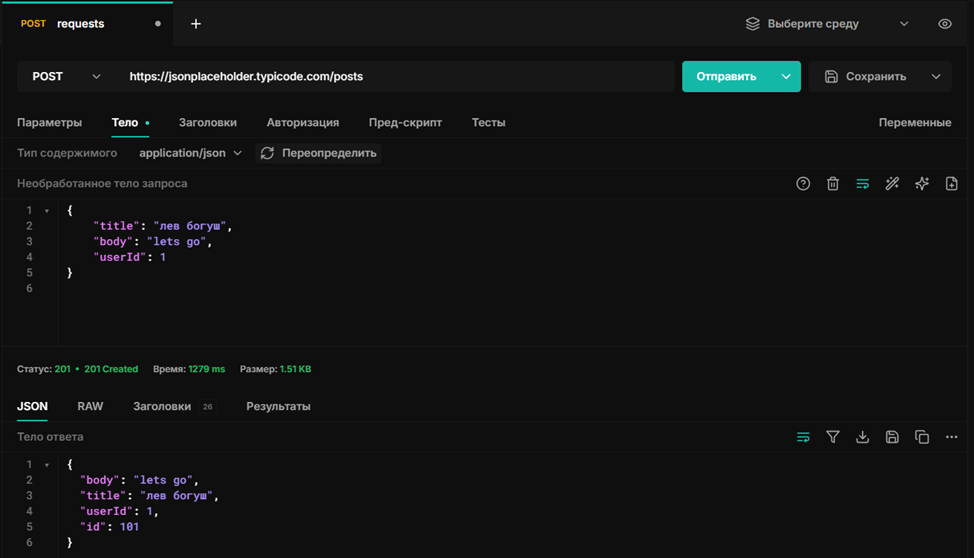
**2. Работа с публичным API:**



Здесь я отправил GET-запрос к URL https://restcountries.com/v3.1/all и получил ответ с статусом 200 OK, что указывает на успешное выполнение запроса. В теле ответа содержится массив объектов, где каждый объект представляет информацию о стране. Структура JSON включает такие данные, как название страны (в формате name, содержащем общепринятое, официальное и местное названия), уникальный идентификатор (id), международные коды страны (cca2, ccn3, cca3) и логическое значение independent, указывающее на независимость страны, и другие данные.

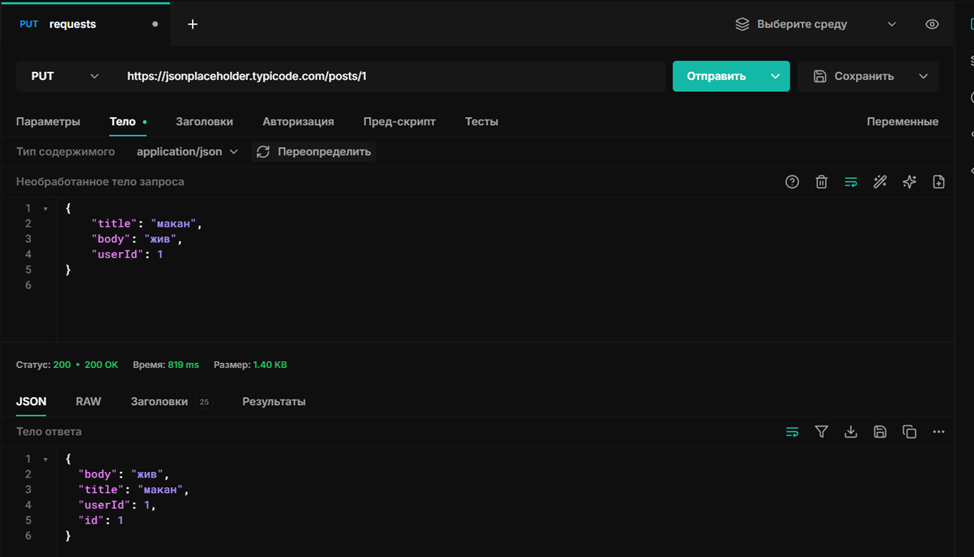
JSON удобен для анализа и обработки данных благодаря своей читаемой структуре и возможности легко передавать сложные данные в виде ключ-значения.

**3. Отправка данных с использованием POST:**



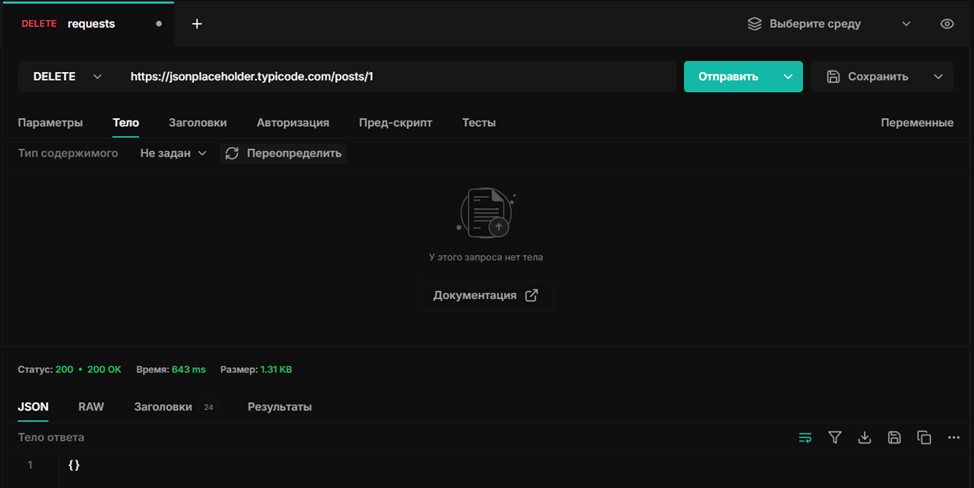
Здесь я отправил POST-запрос к URL <https://jsonplaceholder.typicode.com/posts>с телом запроса в формате JSON, содержащим данные о новом посте: title, body и userId. После отправки запроса получил ответ с статусом 201 Created, что указывает на успешное создание ресурса. В теле ответа содержится информация о созданном посте, включая его уникальный id=101.

**4. Модификация данных с использованием PUT:**



Здесь я отправил PUT-запрос к URL https://jsonplaceholder.typicode.com/posts/1 с телом запроса в формате JSON, содержащим обновленные данные о посте: title, body и userId. После отправки запроса получил ответ с статусом 200 OK, что указывает на успешное обновление ресурса. В теле ответа содержится информация о посте с обновленными данными, включая его уникальный id=1.

**5. Удаление данных с использованием DELETE:**



Здесь я отправил DELETE-запрос к URL https://jsonplaceholder.typicode.com/posts/1 для удаления существующего поста. После отправки запроса получил ответ с статусом 200 OK, что указывает на успешное выполнение операции. Тело ответа пустое, так как при успешном удалении ресурса обычно не возвращается дополнительная информация.

**Заключение:**

В ходе выполнения работы я приобрел практические навыки работы с HTTP-протоколом и API, используя инструмент Postman. Были освоены основные методы HTTP (GET, POST, PUT, DELETE), а также научился создавать запросы, анализировать ответы сервера и работать с публичными API.