

## **Лабораторная работа № 2**

Тема: Основы HTTP.

Цель: Научиться формировать HTTP-запросы, анализировать структуру ответа сервера и использовать разные инструменты для работы с HTTP.

### **Задание:**

#### **1. POST-запрос (форма) в Hoppscotch**

Отправьте запрос POST <https://httpbin.org/post> с параметрами формы:  
username = neo  
password = trinity

Определите, где в ответе отражаются эти параметры.

#### **2. POST-запрос с JSON через curl**

Выполните:

```
curl -v -H "Content-Type: application/json" \
-d '{"task":"lab","status":"done"}' \
https://httpbin.org/post
```

Найдите отправленные данные в блоке "json".

#### **3. Заголовки ответа**

С помощью Hoppscotch выполните запрос GET

<https://httpbin.org/headers>.

Определите: какой User-Agent использует Hoppscotch по умолчанию?



#### **Памятка по curl**

##### **◆ Основные команды**

GET-запрос (по умолчанию):

`curl http://example.com`

GET с параметрами:

`curl "http://httpbin.org/get?user=neo&city=zion"`

POST-запрос (форма):

`curl -d "username=neo&password=trinity" http://httpbin.org/post`

POST с указанием метода явно:

```
curl -X POST -d "key=value" http://httpbin.org/post
```

POST с JSON:

```
curl -H "Content-Type: application/json" \
-d '{"user":"neo","pass":"trinity"}' \
http://httpbin.org/post
```

#### ◆ Полезные флаги

- v — подробный вывод (заголовки запроса и ответа).
- i — показать заголовки ответа вместе с телом.
- o file.html — сохранить ответ в файл.
- L — следовать за редиректами.
- k — игнорировать ошибки SSL (только для тестов!).
- H "Header: value" — добавить свой заголовок.
- X METHOD — указать метод (GET, POST, PUT, DELETE и т.д.).

#### ◆ Примеры

Проверить заголовки сайта:

```
curl -I http://example.com
```

Скачать файл:

```
curl -O http://example.com/file.txt
```

Авторизация:

```
curl -u "user:password" http://example.com/secure
```

#### 📌 Запомни:

По умолчанию curl делает GET.

Для POST используем -d.

Для JSON — -H "Content-Type: application/json".

Всегда полезно добавлять -v для наглядности.