

Практическое занятие № 1

Тема 2. Командные запросы. Протоколы (HTTP, FTP, SMTP) для Curl запросов

Цель работы: Изучить нормативные документы (стандарт), синтаксис и спецификацию API. Изучить методы запросов REST API на сервер.

План занятия.

1. Изучить теоретическую часть.
2. Изучить документацию REST API. Список стандартов. Литература.
3. Выполнить практические задания.
4. Подготовить отчет практической работы.
5. Выполнить задания семинара.
6. Ответить на контрольные вопросы.

1. Теоретическая часть.

Интерфейсом прикладного программирования называют набор исполняемых (runtime) программ или системных вызовов, которые позволяют прикладной программе пользоваться определенной услугой, предоставляемой либо операционной системой, либо другой прикладной программой.

Для функциональной среды открытых систем API (**Application Program Interface**) – это набор программных интерфейсов между прикладными программами (приложениями) и средой.

Спецификации API группируются по основным группам функций (услуг), предоставляемых средой приложениям:

- поддержка пользовательского интерфейса;
- организация процессов обработки данных;
- представление данных для хранения и обмена;
- услуги телекоммуникаций.

Как работать с curl в Windows

Curl (client URL) — это инструмент командной строки на основе библиотеки libcurl для передачи данных с сервера и на сервер при помощи различных протоколов, в том числе: HTTP, HTTPS, FTP, FTPS, IMAP, IMAPS, POP3, POP3S, SMTP и SMTPS.

Он очень популярен в сфере автоматизации и скриптов благодаря широкому диапазону функций и поддерживаемых протоколов.

В практическом занятии показано как можно использовать curl в Windows.

Установка в Windows

Во всех современных версиях Windows, начиная с Windows 10 (версия 1803) и Server 2019, исполняемый файл curl поставляется в комплекте, поэтому ручная установка не требуется.

Для Windows 7 необходимо установить curl (32, 64-bit).

Например, файл: **curl-7.46.0-win64_exe**

Введение в curl и его установка представлена на сайте:

<https://starkovden.github.io/curl-intro-and-installation.html>

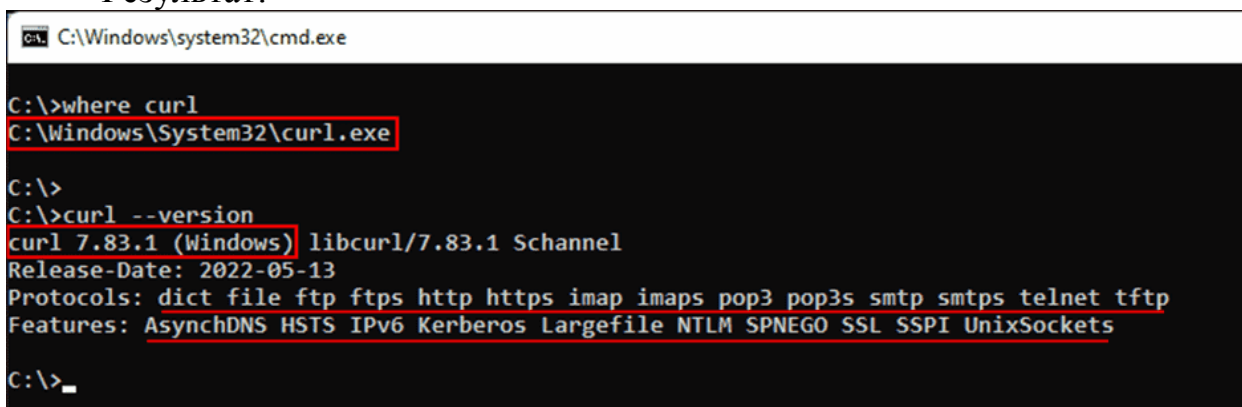
Версия Curl (рис. 1), команда:

```
curl --version
```

Расположение каталога Curl на диске, команда:

```
where curl
```

Результат:



```
C:\Windows\system32\cmd.exe

C:\>where curl
C:\Windows\System32\curl.exe

C:\>
C:\>curl --version
curl 7.83.1 (Windows) libcurl/7.83.1 Schannel
Release-Date: 2022-05-13
Protocols: dict file ftp ftps http https imap imaps pop3 pop3s smtp smtps telnet tftp
Features: AsynchDNS HSTS IPv6 Kerberos Largefile NTLM SPNEGO SSL SSPI UnixSockets

C:\>_
```

Рисунок 1 – Определение местоположения и версии curl в Windows

Команда *curl --version* также выводит список протоколов и функций, поддерживаемых текущей версией curl. Как видно из показанного выше скриншота, к использованию встроенной утилиты curl всё готово. Если вместо этого отображается сообщение об ошибке, curl может быть недоступен потому, что вы используете более раннюю версию Windows (например, Windows 8.1 или Server 2016). В таком случае вам потребуется установить curl в Windows вручную.

Рекомендуется использовать перед вводом новой команды *curl* команду (очистка экрана):

```
cls
```

Синтаксис curl

Команда curl использует следующий синтаксис.
Команда (общий вид):

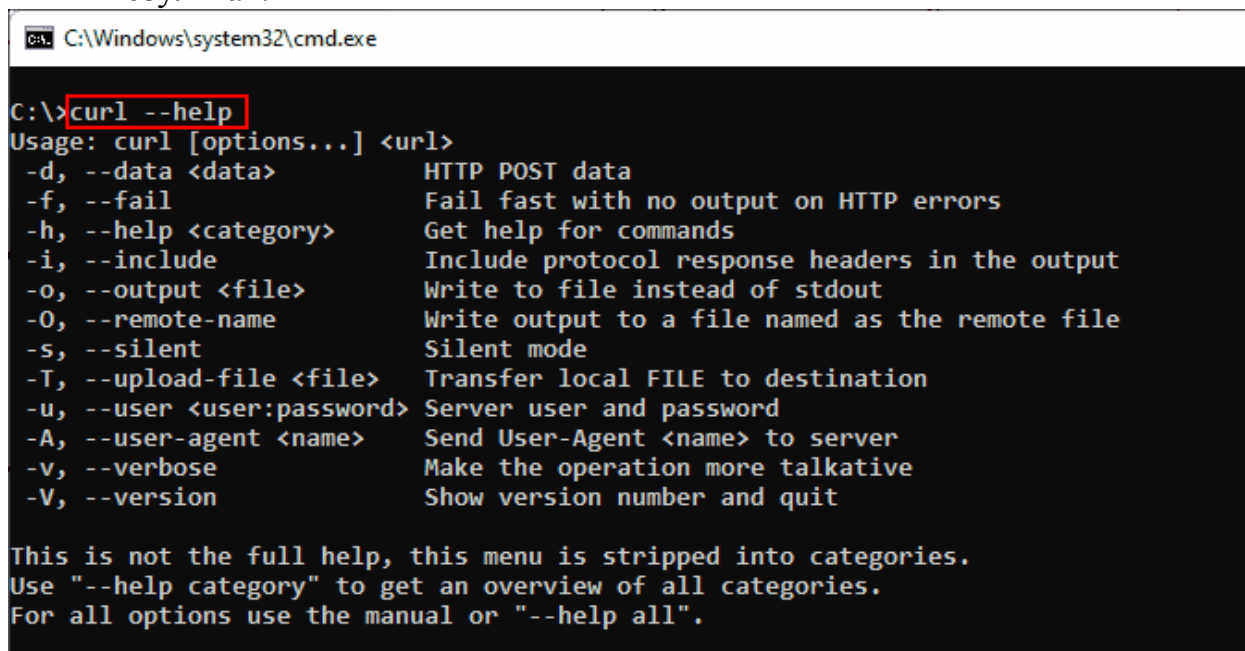
```
curl [options...] [url]
```

Команда (справка):

```
curl --help
```

Инструмент поддерживает различные опции. Как и в любом инструменте командной строки, вы можете использовать для получения справки команду *curl --help* (рис. 2).

Результат:



```
C:\Windows\system32\cmd.exe
C:\> curl --help
Usage: curl [options...] <url>
-d, --data <data>           HTTP POST data
-f, --fail                  Fail fast with no output on HTTP errors
-h, --help <category>      Get help for commands
-i, --include               Include protocol response headers in the output
-o, --output <file>        Write to file instead of stdout
-O, --remote-name           Write output to a file named as the remote file
-s, --silent                Silent mode
-T, --upload-file <file>   Transfer local FILE to destination
-u, --user <user:password> Server user and password
-A, --user-agent <name>    Send User-Agent <name> to server
-v, --verbose               Make the operation more talkative
-V, --version               Show version number and quit

This is not the full help, this menu is stripped into categories.
Use "--help category" to get an overview of all categories.
For all options use the manual or "--help all".
```

Рисунок 2 – Получение справки при помощи команды curl

Для получения подробной справки можно использовать команду *curl --help all*. Справка разделена на категории, поэтому при помощи *curl --help category* можно просмотреть все темы.

Ознакомившись с синтаксисом curl, рассмотрим различные способы применения этого инструмента на примерах.

Основные опции Curl

- # – отображать простой прогресс-бар во время загрузки;
- 0 – использовать протокол http 1.0;
- 1 – использовать протокол шифрования tlsv1;
- 2 – использовать sslv2;
- 3 – использовать sslv3;
- 4 – использовать ipv4;

- b – использовать ipv6;
- A – указать свой USER_AGENT;
- b – сохранить Cookie в файл;
- c – отправить Cookie на сервер из файла;
- C – продолжить загрузку файла с места разрыва или указанного смещения;
- m – максимальное время ожидания ответа от сервера;
- d – отправить данные методом POST;
- D – сохранить заголовки, возвращенные сервером в файл;
- e – задать поле Referer-uri, указывает с какого сайта пришел пользователь;
- E – использовать внешний сертификат SSL;
- f – не выводить сообщения об ошибках;
- F – отправить данные в виде формы;
- G – если эта опция включена, то все данные, указанные в опции -d будут передаваться методом GET;
- H – передать заголовки на сервер;
- I – получать только HTTP заголовок, а все содержимое страницы игнорировать;
- j – прочитать и отправить cookie из файла;
- J – удалить заголовок из запроса;
- L – принимать и обрабатывать перенаправления;
- s – максимальное количество перенаправлений с помощью Location;
- o – выводить контент страницы в файл;
- O – сохранять контент в файл с именем страницы или файла на сервере;
- p – использовать прокси;
- proto – указать протокол, который нужно использовать;
- R – сохранять время последнего изменения удаленного файла;
- s – выводить минимум информации об ошибках;
- S – выводить сообщения об ошибках;
- T – загрузить файл на сервер;
- v – максимально подробный вывод;
- y – минимальная скорость загрузки;
- Y – максимальная скорость загрузки;
- z – скачать файл, только если он был модифицирован позже указанного времени;
- V – вывести версию.

Общие команды curl, связанные с REST

При посещении веб-сайта мы отправляем запрос, используя метод GET. Существуют и другие методы HTTP, которые можно использовать при взаимодействии с REST API.

Общие методы, используемые при работе с конечными точками REST представлены в табл. 1.

Таблица 1 – Методы запросов REST API

HTTP метод	Описание
POST	Создание ресурса
GET	Чтение (получение) ресурса
PUT	Обновление ресурса
DELETE	Удаление ресурса

У curl есть сокращенные имена для различных опций, которые вы включаете в свой запрос.

Вот расшифровка команд:

- **-X GET**. **-X** Показывает, что в запросе будет использоваться какой-то метод. Общие параметры: GET, POST, DELETE, PUT. (Также можете увидеть использование **--get**. Большинство команд curl имеют несколько разных представлений. **-X GET** также можно записать как **--get**);
- **-H** Отправляет пользовательский заголовок. Для каждого заголовка пары ключ-значение включаем дополнительный **-H**.

У curl много возможных команд, но при работе с REST API наиболее распространены следующие (табл. 2).

Таблица 2 – Команды при работе с REST API

curl Команда	Описание	Пример
-i или --include	Включает заголовки ответа в ответ	curl -i http://www.example.com
-d или --data	Включает данные для публикации на URL. Данные должны быть в кодировке URL . Данные также могут быть переданы в теле запроса.	curl -d "data-to-post" http://www.example.com
-H или --header	Отправляет заголовок запроса на ресурс. Заголовки являются общими с запросами REST API, потому что авторизация обычно включается в заголовок	curl -H "key:12345" http://www.example.com

curl Команда	Описание	Пример
-X POST	Задаёт метод HTTP для использования с запросом (в данном примере, POST). При использовании -d в запросе, curl автоматически указывает метод POST. С GET-запросами, в том числе HTTP-метод является необязательным, поскольку GET-метод используется по умолчанию	curl -X POST -d "resource-to-update" http://www.example.com
@filename	Загружает контент из файла	curl -X POST -d @mypet.json http://www.example.com

Общие команды curl

Для разминки вот основные команды/синтаксис curl с различными опциями и их пояснениями (табл. 3).

Таблица 3 – Основные команды/синтаксис curl

КОМАНДА	ОБЪЯСНЕНИЕ
curl http://example.com	Извлекает содержимое веб-страницы. (Пример без параметров)
curl -o filename.html http://example.com	Загружает содержимое веб-страницы в указанный файл.
curl -I http://example.com	Извлекает только заголовки HTTP из URL.
curl -L http://example.com	Отслеживает HTTP-перенаправления, что полезно для определения конечного пункта назначения URL-адреса с несколькими перенаправлениями.
curl -u username:password http://example.com	Выполняет запрос с HTTP-аутентификацией.
curl -x http://proxyserver:port http://example.com	Использует прокси-сервер для запроса.
curl -d "param1=value1&param2=value2" -X	Отправляет POST-запрос с данными на сервер.

<code>POST http://example.com</code>	
<code>curl -H "X-Custom-Header: value" http://example.com</code>	Добавляет пользовательский заголовок к запросу.
<code>curl -s http://example.com</code>	Заставляет curl работать в тихом режиме, подавляя индикатор прогресса и сообщения об ошибках.
<code>curl -X PUT -T file.txt http://example.com</code>	Загружает файл на сервер методом PUT.
<code>curl -A "User-Agent-String" http://example.com</code>	Имитирует пользовательский агент, отправляя пользовательский заголовок User-Agent.
<code>curl --json '{"tool": "curl"}' https://example.com/</code>	Отправить данные JSON на cURL

Используйте `curl -help` для просмотра дополнительных команд.

Эти команды демонстрируют некоторые из основных функций `curl` взаимодействия с веб-серверами, API и обработки различных методов HTTP и типов данных.

Выгрузка файла

Опция `--upload-file` (или `-T`) позволяет выгружать локальный файл на удалённый сервер. Показанная ниже команда выгружает файл из локальной системы на удалённый веб-сервер по протоколу FTPS:

Командная строка:

```
curl -kT C:\Users\Surender\Downloads\sample1.zip -u testlab\surender https://192.168.0.80/awesomewebsite.com/files/
```

Результат:

```

C:\Windows\system32\cmd.exe
C:\>curl -kT C:\Users\Surender\Downloads\sample1.zip -u testlab\surender https://192.168.0.80/awesomewebsite.com/files/
Enter host password for user 'testlab\surender':
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 10.0M 0 0 100 10.0M 0 10.1M --:--:-- --:--:-- --:--:-- 10.1M
C:\>curl -kT https://awesomewebsite.com/files/sample1.zip
HTTP/1.1 200 OK
Content-Length: 10525461
Content-Type: application/x-zip-compressed
Last-Modified: Sat, 08 Oct 2022 09:07:09 GMT
Accept-Ranges: bytes
ETag: "5789b358f5dad81:0"
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Sat, 08 Oct 2022 09:14:28 GMT

```

Рисунок 3 – Выгрузка файла на удалённый сервер

Опция -k добавляется для устранения проблем с сертификатами на случай, если веб-сервер использует самоподписанный сертификат. Наклонная черта в конце URL сообщает curl, что конечная точка является папкой. Можно указать несколько имён файлов, например "{sample1.zip,sample2.zip}". Ниже показано, как с помощью одной команды curl можно выгрузить на сервер несколько файлов:

Командная строка:

```
curl -kT sample[1-5].zip -u testlab\surender  
https://192.168.0.80/awesomewebsite.com/files/
```

Результат:

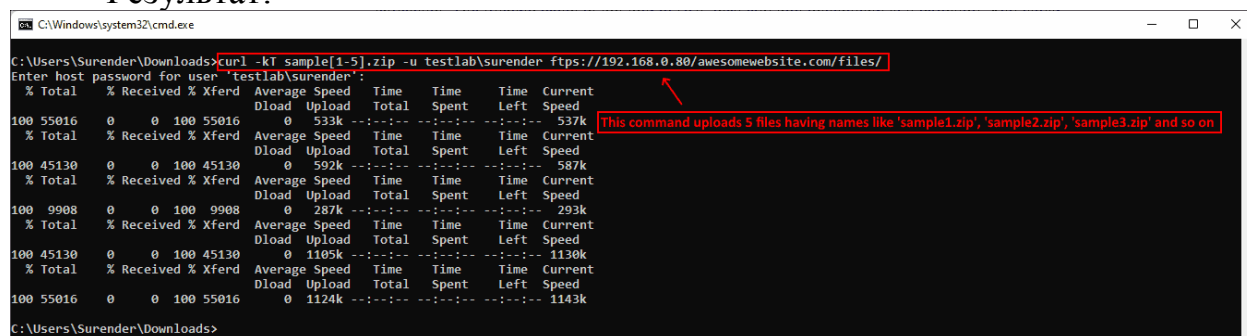


Рисунок 4 – Выгрузка нескольких файлов на сервер

1 Отправка электронного письма

Так как curl поддерживает протокол SMTP, его можно использовать для отправки электронного письма. Показанная ниже команда позволяет отправить электронное письмо при помощи curl:

Командная строка:

```
curl --insecure --ssl-reqd smtps://mail.yourdomain.com  
--mail-from sender@yourdomain.com --mail-rcpt  
receiver@company.com --user sender@yourdomain.com --  
upload-file email_msg.txt
```

Результат:

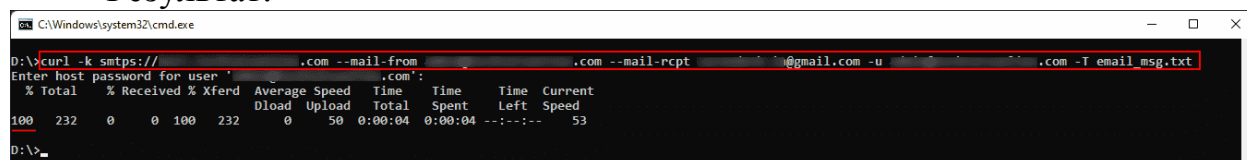


Рисунок 5 – Отправка электронного письма командой curl

2.Документация REST API

1.Документация REST API.

<https://starkovden.github.io/intro-rest-api.html>

Список стандартов.

RFC 2616 обязательно для чтения всем, кто хочет понять протокол HTTP.

<https://datatracker.ietf.org/doc/html/rfc2616>

RFC 2396 объясняет синтаксис URL.

RFC 2109 определяет работу cookies.

RFC 1867 определяет формат File Upload Post.

Литература

Учебное пособие (2024 г)

2.Everything curl.Daniel Stenberg and friends. March 18th, 2024. 551 pages

<https://daniel.haxx.se/everything-curl/everything-curl.pdf>

Format book: PDF (3,61 MB), ePUB (3,36 MB), HTML (2,58 MB)

<https://daniel.haxx.se/everything-curl/>

3.The book contents is rendered by mdBook

<https://github.com/rust-lang/mdBook>

4.Github

<https://everything.curl.dev/source/index.html>

3.Практические задания

Цель: Изучить командные запросы Curl, основные опции Curl. Изучить методы запросов REST API на сервер. Научиться выполнять командные запросы, на сервер, используя команду Curl в операционной системе Windows.

Для запуска командной строки Windows используйте командную строку:

```
cmd
```

нажмите Enter.

Далее, можно использовать Curl (client URL).

Чтобы определить местоположение *curl* и его версию в системе, можно использовать следующие команды и результат (рис. 1).

Все команды *curl* запустить с диска C. Чтобы перейти на диск C, наберите команду.

Переход на диск C: осуществляется командой:

```
cd c:\ или cd\
```

Задание № 1. Выполнить запрос на сервер в curl.

| HTTP-запрос GET

- 1.Выполните запрос, на сервер, используя метод GET протокола http.
- 2.В результате запросов вы должны получить код ответа сервера, например: 100, 200, 303, 400, 500.
- 3.Результат кода ответа от сервера методом GET записать в таблицу 5.

Для запроса на сервер и получения кода ответа с сервера используйте список web-адресов крупнейших российских интернет-магазинов (рейтинг ТОП-100).

URL: <https://top100.datainsight.ru/>

Для запроса на сервер можно составить собственный список web-адресов сайтов.

Количество кодов ответов от сервера рекомендуется взять по каждому диапазону, представленного в таблице 4.

Список кодов состояния HTTP

1.URL:https://ru.wikipedia.org/wiki/Список_кодов_состояния_HTTP

2.URL: <https://hostzealot.ru/blog/about-servers/kody-otvetov-servera-polnyi-spisok>

Таблица 4 – Диапазон кодов ответов сервера

Диапазон	Расшифровка диапазона в спецификации протокола	Описание
100÷199	Informational	Информационный
200÷299	Successful	Запрос клиента успешно принят и обработан
300÷399	Redirection	Запрос клиента переадресован
400÷499	Client Error	Ошибка клиента
500÷599	Server Error	Ошибка сервера

Таблица 5 – Результат запроса на сервер, методом GET

Код ответа	Описание

При использовании curl с URL адресом и без указания опций запрос по умолчанию использует метод GET протокола HTTP.

Синтаксис, пример запроса:

```
curl -I https://*****.**
```

Команда, например:

```
curl www.google.com
```

Получение заголовков HTTP

Если вы хотите посмотреть, какие заголовки отдает сервер, то можно использовать опции -I или -head. Они позволяют получить заголовок без тела документа.

Команда, например:

```
curl -I https://www.google.com
```

Результат:

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
Date: Thu, 04 Jun 2020 15:07:42 GMT
Server: gws
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked
Expires: Thu, 04 Jun 2020 15:07:42 GMT
Cache-Control: private
Set-Cookie: 1P_JAR=2020-06-04-15; expires=Sat, 04-Jul-2020 15:07:42 GMT; path=/;
domain=.google.com; Secure
Set-Cookie: <cookie_info>
```

Выполните команды:

```
curl https://4sysops.com
```

Приведённая выше команда, по сути эквивалентна `curl --request GET 4sysops.com`, отправляющей запрос GET к 4sysops.com по протоколу HTTPS. Чтобы указать версию протокола HTTP (например, http/2), используйте опцию --http2:

```
curl --http2 https://4sysops.com
```

В случае URL, начинающегося с HTTPS, curl сначала пытается установить соединение http/2 и автоматически откатывается к http/1.1, если это не удаётся.

Также он поддерживает другие методы, например, HEAD, POST, PUT и DELETE. Для использования этих методов вместе с командой curl нужно указать опцию `--request` (или `-X`), за которой следует указание метода. Стоит заметить, что список доступных методов зависит от используемого протокола.

Метод GET используется для получения данных с сервера. Это метод предназначен только для чтения, поэтому риск изменения или повреждения данных отсутствует.

Для просмотра параметров запроса сервера с применением метода GET (Request Method) необходимо открыть вкладку браузера клавишами **CTR + U** или **F12**.

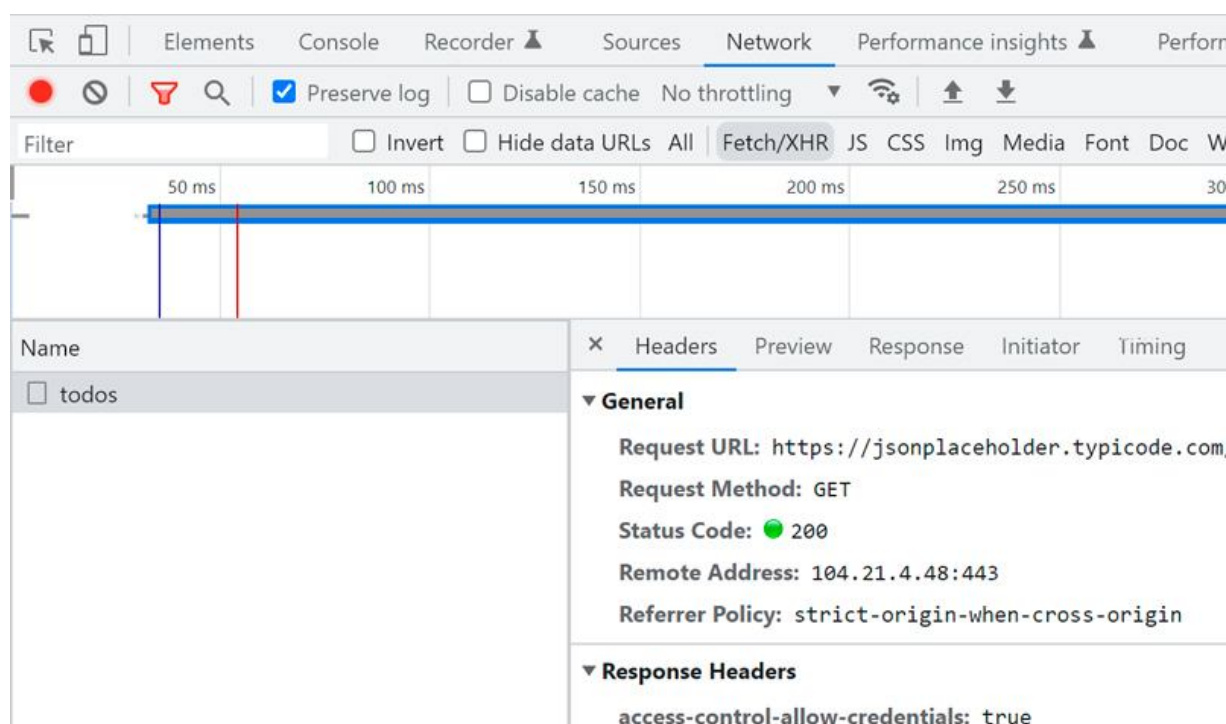


Рисунок 6 – Основные параметры метода GET

Задание № 2. Выполнить запрос на сервер в curl.

HTTP-запрос POST

1. Выполните запрос, на сервер, используя метод POST (до 5 серверов).

Синтаксис, пример запроса:

```
curl -X POST https://*****.**
```

Пример запроса:

```
curl -X POST https://httpbin.org/anything
```

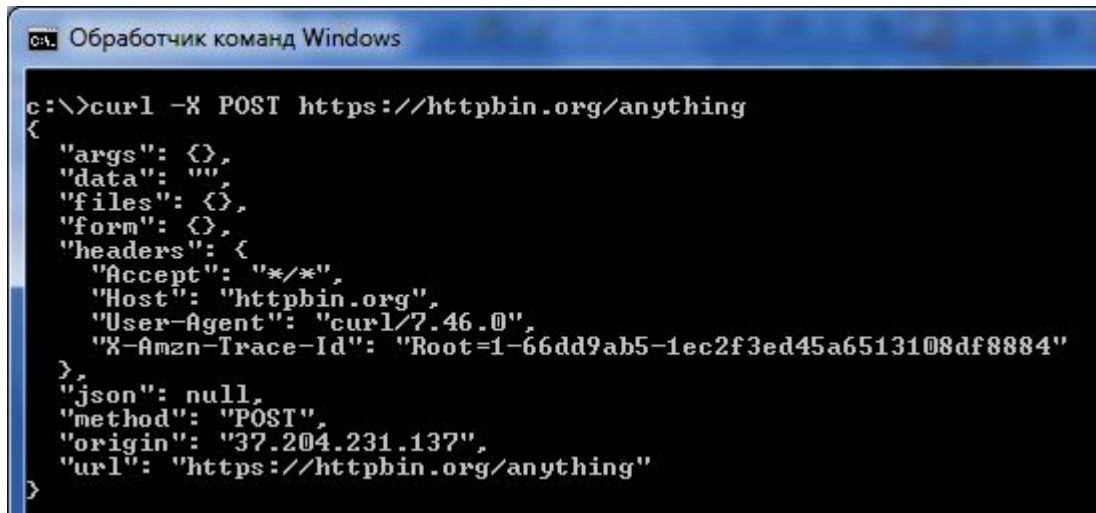


Рисунок 7 – Результат запроса

2. Выполните запрос, на сервер, используя метод POST (формат JSON)

curl позволяет задать заголовки для HTTP-запроса с помощью флага *-H*.

Например, если вы отправляете JSON в POST-запросе, следующая команда *curl* показывает, как вы можете установить заголовок Content-Type для запроса:

Пример запроса:

```
curl -X POST -H 'Content-Type: application/json' -d '{}'
```

Флаг *-H* также может быть указан с помощью более длинного флага *--header*. В этом случае ваша команда *curl* будет выглядеть следующим образом:

```
curl -X POST --header 'Content-Type: application/json' -d '{}'
https://httpbin.org/anything
```

3. Показать заголовок запроса и ответа.

Для получения ожидаемого заголовка запроса и ответа, используйте следующую команду:

```
curl -v https://www.dzen.ru
```

6. Открыть только заголовок ответа

Проверить заголовок ответа, то для этого можно использовать следующий синтаксис.

```
curl --head https://www.dzen.ru
```

4.Отчет практической работы

Подготовить отчет практической работы в соответствии со структурой отчета. Отчет представить в формате А4, файл Word.

Структура отчета

1. Титульный лист.
2. Тема. Цель работы. Задание.
3. Решение (программное обеспечение, результат запроса и др.).
5. Выводы.
6. Список литературы.
7. Приложение (при необходимости).

Отчет, представить на проверку:

Папка: Иванов АА (ПИ21-1)/ПР-2/

Файл (word): **ИвановАА-пр2.docx**

5.Семинар № 1

Тема: Командные запросы. Протоколы (HTTP, FTP, SMTP) для Curl запросов

Цель: Организация занятий с возможностью изучения и применения новых, актуальных методов, технических и технологических решений, решения проблем, перспектив для повышения эффективности технологии автоматизации открытых данных.

Структура семинара (План занятий).

- 1.Организация (проверка присутствующих, готовность обучающихся к занятию, формирование групп – 2-3 чел. на одну тему).
- 2.Ознакомление студентами с темами семинара.
- 3.Формирование групп и выбор темы семинара (формирование списка групп семинара и тем).
- 4.Выполнить задание семинара:
 - поиск информации по тематике;
 - составить рабочий проект презентации (согласовать с преподавателем);

- подготовить презентацию и доклад по тематике семинара;
 - выступление (2-3 мин/группа). Ответы на вопросы.
5. Заключение. Подведение итогов семинара.

Темы семинара

1. Запросы на сервер методом GET, привести примеры.
2. Запросы на сервер методом POST, привести примеры.
3. Командные запросы через командную строку, с использованием утилиты для передачи данных через различные протоколы (HTTP, FTP, SMTP), привести примеры.

Источники информации

1. Статьи (Периодические публикации, Журналы РФ, зарубежные) 2-3 статьи.
2. Практические материалы сценариев (примеры) с официальных и частных сайтов.

6. Контрольные вопросы

1. Какие методы запросов на сайт протокола HTTP с применением REST API выполняются.
2. Какой метод запроса на сервер использует curl по умолчанию.
3. Какой метод запроса REST API позволяет читать ресурсы.
4. Какая команда REST API позволяет извлекать содержимое веб-страницы.
5. Какая команда REST API позволяет извлекать только заголовки HTTP из URL.