

Практическое занятие № 3

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

Curl запросы для тестирования и проверки API. Curl запросы для получения доступа к удаленному ресурсу. Curl запросы для мониторинга работоспособности сайтов и веб-приложений. Настройка Curl запросов под конкретные потребности.

Цель: Изучить командные запросы Curl. Научиться выполнять командные запросы, на сервер, используя команду Curl в операционной системе Windows.

План занятия.

- 1.Выполнить практические задания.
- 2.Подготовить отчет практической работы.
- 3.Ответить на контрольные вопросы.

Литература

Учебное пособие (2024 г)

- 1.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/>

- 2.The book contents is rendered by mdBook
<https://github.com/rust-lang/mdBook>

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

1.Практические задания

Задание № 1. Используем curl для получения веб-страницы

- 1.Выполнить командный запрос, используя curl для извлечения HTML-код веб-ресурса.

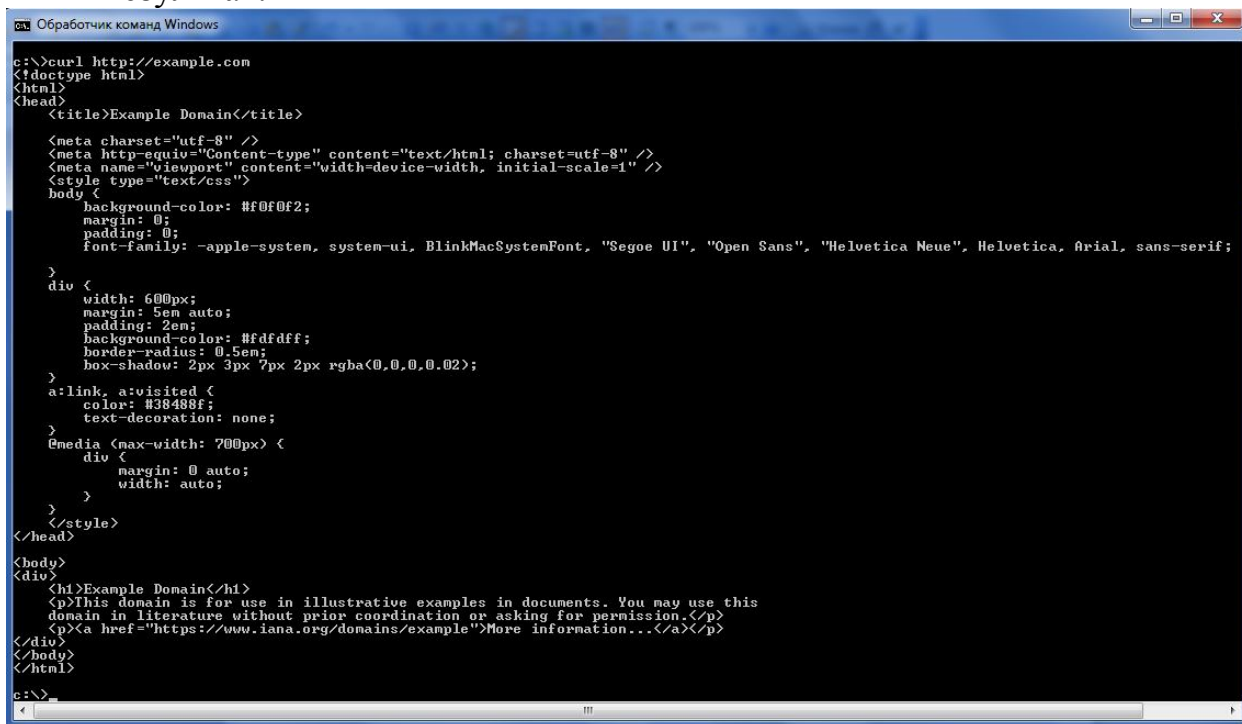
REST следует той же модели, что и веб. При вводе http-адреса в адресную строку браузера мы говорим браузеру, что он должен сделать HTTP-запрос к ресурсу на сервере. Сервер возвращает ответ, а наш браузер преобразует ответ в более визуальное отображение.

Чтобы можно увидеть, например, как curl извлекает HTML-код веб-ресурса. Откройте терминал и введите следующее (рис. 1).

Команда:

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

Результат:



```
c:\>curl http://example.com
<!doctype html>
<html>
<head>
  <title>Example Domain</title>

  <meta charset="utf-8" />
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <style type="text/css">
    body {
      background-color: #f0f0f2;
      margin: 0;
      padding: 0;
      font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
    }
    <div>
      width: 600px;
      margin: 5em auto;
      padding: 2em;
      background-color: #fdfdff;
      border-radius: 0.5em;
      box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
    a:link, a:visited {
      color: #38488f;
      text-decoration: none;
    }
    @media (max-width: 700px) {
      <div>
        margin: 0 auto;
        width: auto;
      }
    }
  </style>
</head>
<body>
<div>
  <h1>Example Domain</h1>
  <p>This domain is for use in illustrative examples in documents. You may use this
  domain in literature without prior coordination or asking for permission.</p>
  <p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>
c:\>
```

Рис. 1 – Curl извлекает HTML-код веб-ресурса

Curl извлечет HTML-код для сайта example.com. Задача браузера - сделать этот код визуально читаемым. curl показывает, что мы на самом деле получаем.

Задание № 2. Запросы и ответы содержат заголовки

Метод GET

1.Выполнить командный запрос, используя curl для получения информации о запросе и ответе только заголовка методом GET.

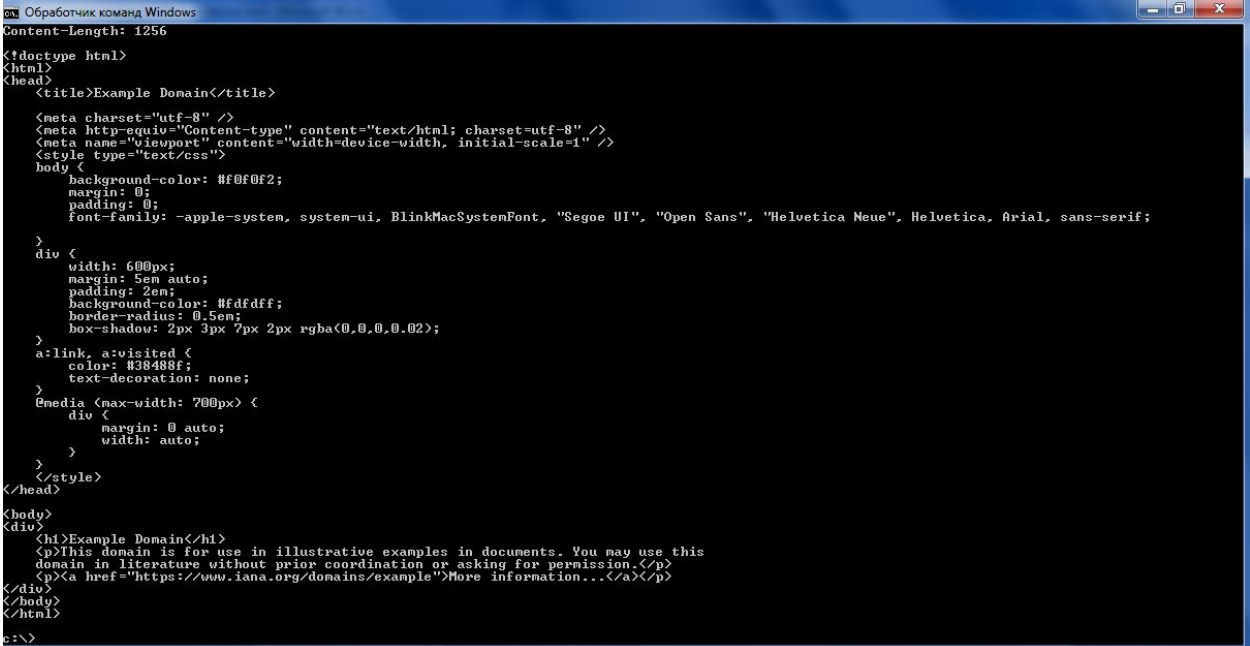
При вводе адреса веб-сайта мы видим только текст ответа. Но на самом деле происходит гораздо больше процессов. Когда мы делаем запрос, мы отправляем заголовок запроса, который содержит информацию о запросе. Ответ также содержит заголовок ответа.

Для того чтобы увидеть заголовок ответа в запрос curl, добавим ключ - i (рис. 2).

Команда:

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

Результат:



```
Обработчик команд Windows
Content-Length: 1256
<!doctype html>
<html>
<head>
  <title>Example Domain</title>
  <meta charset="utf-8" />
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <style type="text/css">
    body {
      background-color: #f0f0f2;
      margin: 0;
      padding: 0;
      font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
    }
    div {
      width: 600px;
      margin: 5em auto;
      padding: 2em;
      background-color: #fdfdff;
      border-radius: 0.5em;
      box-shadow: 2px 3px 7px rgba(0,0,0,0.02);
    }
    a:link, a:visited {
      color: #38488f;
      text-decoration: none;
    }
    @media (max-width: 700px) {
      div {
        margin: 0 auto;
        width: auto;
      }
    }
  </style>
</head>
<body>
<div>
  <h1>Example Domain</h1>
  <p>This domain is for use in illustrative examples in documents. You may use this
  domain in literature without prior coordination or asking for permission.</p>
  <p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>
c:\>
```

Рис. 2 – Информация о запросе и заголовок ответа

Чтобы в ответе получить только заголовок, используйте ключ `-I` (рис. 3).

Команда:

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

Результат:



```
Обработчик команд Windows
c:\>curl http://example.com -I
HTTP/1.1 200 OK
Accept-Ranges: bytes
Age: 360642
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Fri, 26 Jul 2024 15:18:33 GMT
Etag: "3147526947"
Expires: Fri, 02 Aug 2024 15:18:33 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECACC (nyd/D11B)
X-Cache: HIT
Content-Length: 1256
c:\>_
```

Рис. 3 – Ответ получил только заголовок

Уточним метод. Метод GET (чтение) подразумевается по умолчанию, когда не указан другой метод, но мы сделаем это здесь явно с параметром `-X`: (рис. 4).

Команда:

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

Результат:

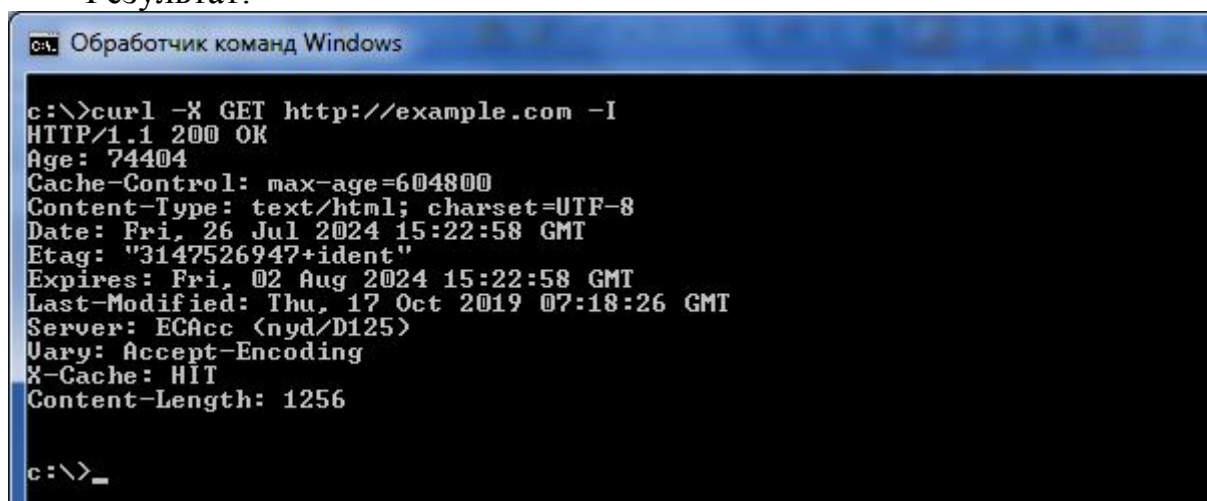
A screenshot of a Windows Command Prompt window titled "Обработчик команд Windows". The command prompt shows the execution of the command `c:\>curl -X GET http://example.com -I`. The output is as follows:
HTTP/1.1 200 OK
Age: 74404
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Fri, 26 Jul 2024 15:22:58 GMT
Etag: "3147526947+ident"
Expires: Fri, 02 Aug 2024 15:22:58 GMT
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
Server: ECACC <nyd/D125>
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1256
The prompt ends with `c:\>_`.

Рис. 4 – Метод GET (чтение)

Задание № 3. Количество секунд, которое заняло обращение к сайту

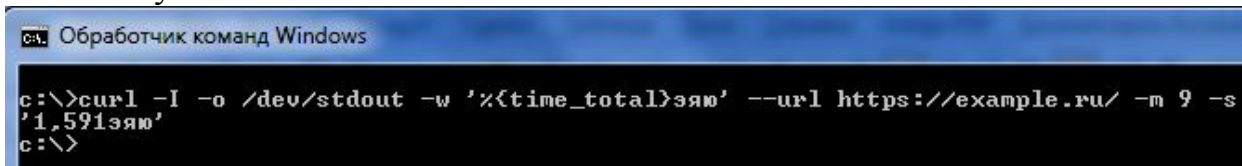
1. Выполнить командный запрос, используя `curl` для получения оценки времени обращения к сайту (рис. 5).
2. Выполните командные запросы с 5 разных сайтов.
3. Оцените интервал времени, которое заняло обращение к сайту.

`-I` значит, что мы запрашиваем не всю страницу, а только её «заголовки HTTP». Нам не нужен каждый раз текст страницы целиком, чтобы убедиться в работоспособности сайта. Поскольку мы будем проверять сайт часто, важно сделать размер передаваемых данных как можно меньше. Это экономит и трафик, и электричество, и нагрузку на сервер — и на живую природу.

Команда:

```
curl -I -o /dev/stdout -w '%{time_total}эяю' --url  
https://example.ru/ -m 9 -s
```

Результат:



```
с-ч. Обработчик команд Windows
c:\>curl -I -o /dev/stdout -w '%{time_total}эяю' --url https://example.ru/ -m 9 -s
'1.591эяю'
c:\>
```

Рис. 5 – Количество секунд, которое заняло обращение к сайту

Обратите внимание, сколько у вас запасного трафика (не используется за месяц) на хостинге. Ниже увидите, какой объём данных передаётся и сможете посчитать, хватит ли вам запасов; если что, период проверки сайта можно увеличить до 5, 10, или даже 30-60 минут. Хотя в этом случае картина будет не такой полной — и мелкие перебои могут остаться незамеченными; но, в целом, при мониторинге сайта важнее обнаруживать длительные проблемы, чем однократные перебои. Также, какой трафик вы можете себе позволить на вызывающем компьютере? В моём случае — настольный ПК с безлимитным трафиком — это не так важно; но для мобильного устройства или тарифа с лимитами стоит посчитать и, возможно, увеличить интервалы проверки.

Далее: `-o /dev/stdout` значит «ответ, полученный Курлом от сервера, направить в такой-то файл». И в данном случае это не файл на диске, а `/dev/stdout` — Стандартное устройство вывода. Обычно «Стандартное устройство вывода» — это наш экран, где мы можем видеть итоги работы программы. Но в скрипте мы часто направляем этот «стандартный вывод» на дальнейшую обработку (сейчас — сохраняем в переменную `info`). И дальше мы будем, чаще всего, либо направлять итоги команд в переменные, либо передавать их следующим командам по цепочке. Для создания цепочек из команд служит оператор `pipe` («Труба»), изображаемый как вертикальная черта (`"|"`).

`-w '%{time_total}эяю'`: здесь `-w` значит «сформатировать и выдать на стандартное устройство вывода такую-то дополнительную информацию». А конкретно, нас интересует `time_total` — сколько всего времени заняла передача запрос-ответа между нами и сервером. Вы, наверное, знаете, что есть более простая команда, `ping` («Пинг»), чтобы быстро запросить сервер и получить от него ответ, «понг». Но Пинг только проверяет, что сервер хостинга жив, и сигнал туда-обратно идёт столько-то времени. Это показывает предельную скорость доступа, но ещё ничего не говорит нам о том, как быстро сайт выдаёт реальное содержимое. Пинг может работать быстро, и при этом сайт может тормозить или вообще не работать — из-за высокой нагрузки или каких-то внутренних проблем. Поэтому мы используем именно Курла и получаем действительное время выдачи сервером нашего содержимого.

(И по этому параметру можно судить, достаточно ли сервер производительно работает для ваших задач, удобно ли пользователям его типичное время отклика. Не возникает ли проблем — например, мои сайты

на многих хостингах с течением времени замедлялись, и приходилось искать другой хостинг).

Вы заметили странные буквочки «эяю» после {time_total}? Их добавили как уникальную метку, которой наверняка не будет в отправленных нам с сервера заголовках (хотя делать предположения, что будет и чего не будет у пользователей ваших программ — дурной тон и дорога в бездну).

По этой метке (надеюсь) мы потом сможем легко вытащить нужный нам кусочек информации из целой кучи строк в переменной info. Итак, curl -w '%{time_total}эяю' (при правильных остальных параметрах) выдаст нам:

0.215238эяю

Это количество секунд, которое заняло обращение к сайту, плюс наша метка. (Кроме этого параметра, в переменной info нас будет интересовать, главным образом, «Status code» — код состояния, или, по-простому, код ответа сервера. Обычно, когда сервер выдаёт запрошенный файл, код «200». Когда страница не найдена, это знаменитое 404. Когда на сервере ошибки, это, чаще всего, 500 с чем-то).

Задание № 4. Способы объединения данных (Merge, BI)

1. Изучить способы объединения (Merge) данных из разных источников.
2. Опишите способы, инструменты, приведите примеры объединения данных.
3. Укажите программы для объединения данных и составьте рекомендации для повышения эффективности консолидированной базы данных.

Источники информации.

Консолидация баз данных: этапы, методы и примеры

URL: <https://habr.com/ru/companies/ruvds/articles/732006/>

Объединение данных Yandex DataLens

URL:

https://yandex.cloud/ru/docs/datalens/concepts/data-join?utm_referrer=https%3A%2F%2Fya.ru%2F

2. Изучите использование инструментов Business Intelligence (BI).

Опишите этапы выполнения работы с инструментами, приведите примеры.

Business Intelligence и бизнес-аналитика: стратегия, этапы, процессы и инструменты.

URL: <https://habr.com/ru/articles/714910/>

2.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**

3. Контрольные вопросы

- 1.Какая команда позволяет извлекать HTML-код веб-ресурса.
- 2.Какая команда позволяет увидеть заголовок ответа.
- 3.Что выдает команда –X в результате запроса.
- 4.Какая команда позволяет выдать только заголовки веб-страницы.
- 5.Какая команда позволяет определить время на обращение к сайту.
- 6.Назовите способы объединения данных.
- 7.Назовите инструменты для объединения данных.
- 8.Для какой цели применяют инструмент Business Intelligence.
- 9.Какие учитывают факторы при выборе технологий для консолидации баз данных.
- 10.Какие показатели отслеживает программное обеспечение при мониторинге и оптимизации консолидированной базы данных.