HTTP

https://www.youtube.com/embed/PpdQQjPS0MA

Структура HTTP-запросов и ответов

В HTTP и запрос, и ответ имеют похожую структуру:

1. URL
2. Метод
3. Версия HTTP
4. Заголовки
5. Статус-код (обязательно только для HTTP-ответов)
6. Тело (необязательно)

Пример HTTP-запроса:

```
POST /cgi-bin/process.cgi HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

licenseID=string&content=string&/paramsXML=string
```

, где

- `POST` — метод,
- `/cgi-bin/process.cgi` — URL,
- `HTTP/1.1` — версия протокола HTTP,
- `User-Agent`, `Host` и другие — заголовки

- `licenseID=string&content=string&/paramsXML=string` — тело запроса.

Пример HTTP-ответа:

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
Content-Length: 88
Content-Type: text/html
Connection: Closed

<html>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

В данном ответе отстутствует URL и метод, зато появился статус-код `200`, означающий "успех".

1. URL запроса

Структура URL-адреса

```
http://blog.example.com:80/catalog/category/knifes?key1=value1&key2=value2#anchor
```

где

- **http://** — протокол
- **blog** — поддомен (субдомен, домен третьего уровня)
- **example** — домен (домен второго уровня)
- **com** — доменная зона (домен первого уровня)
- **80** — порт. для http 80, для https 443
- **/catalog/category/knifes** — ЧПУ-адрес (человеко-понятный URL)
- **key1=value1&key2=value2** — GET параметры запроса
- **anchor** — якорь

Абсолютные и относительные URL

Абсолютные адреса в проекте - зло. Используйте относительные.

Абсолютный URL

```
https://developer.mozilla.org/ru/docs/Learn
```

Абсолютный URL, скрыт протокол

```
//developer.mozilla.org/ru/docs/Learn
```

Относительный URL (неправильно) □□

```
ru/docs/Learn
```

если мы на странице https://domain.ru/ru/about, то преобразуется в https://domain.ru/ru/aboutru/docs/Learn

Относительный URL (правильно) □

```
/ru/docs/Learn
```

если мы на странице https://domain.ru/ru/about, то преобразуется в https://domain.ru/ru/docs/Learn

Якоря (анкоры) ⚓

Якоря придуманы в HTML, чтобы давать ссылки не просто на HTML страницу, а на определенное место на странице. Позже якоря стали активно использоваться для передачи параметров в JS.

URL с якорем

```
https://domain.ru/url#someAnchor
```

HTML-элемент, на который ведет якорь

```
<h2 name="someAnchor">Заголовок с якорем</h2>
```

2. Методы HTTP-запросов

Если при повторении одного и того же запроса на сервере ничего не меняется, он считается идемпотентным. Если повторение одинакового запроса несколько раз изменяет состояние сервере, то он **не** идемпотентен и, вообще, говоря менее безопасен.

Метод	POST-параметры	Идемпотентность	Предназначение
GET	нет	да	получает информацию о ресурсе
POST	да	нет	создает ресурс
PUT	да	да	заменяет ресурс целиком
PATCH	да	да	редактирует ресурс
DELETE	нет	да	удаляет ресурс
HEAD	нет	да	не получает body, только отправляет и получает HTTP-заголовки

PUT заменяет сущетвующую сущность. Те элементы сущности, которые не представлены в запросе, будут очищены или заменены на null.

GET и POST параметры

В контексте методов HTTP-запросов стоит уопмянуть про GET и POST параметры, так как это вы будете постоянно с ними встречаться и так проще понять, как используются методы.

GET-параметры передаются в URL запроса. Пример URL с GET-параметрами я приводил выше. Любой из перечисленных ниже методов может иметь или не иметь GET-параметры, это совершенно нормально.

POST-параметры это любые параметры, передаваемые в теле (body) запроса. POST-параметры не следует использовать для некоторых методов, например для **GET**, **DELETE** и **HEAD**.

3. Статус-коды ответа

По группам

- **100 - 199** Информационные
- **200 - 299** Успешные
- **300 - 399** Редиректы
- **400 - 499** Клиентские ошибки
- **500 - 599** Серверные ошибки

Популярные коды ответа

- Для редиректов SEO-шники рекомендуют **301** редиректы.
- **401** не авторизован (юзер не представился)
- **403** запрещено (юзер не уполномочен)
- **404** ресурс не найден
- **418** я чайник
- **419** кука протухла (обычно ошибка проверки CSRF)
- **502** Bad Gateway. Программа веб-сервер (Nginx) получила неправильный ответ от бэкенда (PHP). Скорее всего, что то сломалось или неправильно настроено на сервере
- **504** Gateway Timeout. Программа веб-сервер устала ждать ответа от бэкенда и отменила запрос к нему. Скорее всего, на бэкенде происходит что то очень долгое, какое-то большое вычисление. Ну и все равно это не нормально, такого быть не должно.
- **503** Maintenance Mode. На сервере ведутся технические работы.

Коды ответа, как и заголовки, можно посмотреть в браузере на вкладке Network или получить командой `curl -I domain.ru`. В отличие от заголовков, коды ответа бывают только в ответе и не используются в запросе.

4. Заголовки HTTP

Заголовки — часть HTTP-запроса и/или HTTP-ответа, невидимые пользователю. Представляют собой пары **ключ: значение**. Заголовки можно посмотреть в браузере на вкладке Network или получить командой `curl -I domain.ru` (так можно получить только заголовки **ответа** сервера).

Заголовки запроса

- **Host** — ВАЖНО. домен, по которому мы переходим
- **Cookie** — куки, отдаваемые браузером серверу
- **User-Agent** — браузер, которым делаем запрос
- **Accept-Language** и **Accept-Encoding** — принимаемые браузером язык и кодировка
- **Referer** — предыдущая страница
- **Authorization** — реквизиты для базовой авторизации (логин пароль)

Заголовки ответа

- **Cache-Control** — сервер говорит браузеру как кэшировать данные
- **Content-Type** — тип и подтип содержимого, а также кодировка и приложение для открытия содержимого. Примеры:

    - **Content-Type: text/html; charset=UTF-8** — Тип текст, подтип HTML. Кодировка UTF-8
    - **Content-Type: image/gif** — Тип картинка, подтип GIF
    - **Content-Type: application/pdf** — Тип "открываемый внешним приложением", подтип PDF
- **Content-Disposition** — говорит браузеру скачивать или открывать документ как веб-страницу
- **Location** — заголовок для редиректа
- **Set-Cookie** - сервер передает браузеру куку
- **WWW-Authenticate** — выводит окно с базовой аутентификацией
- **Content-Encoding** — сервер говорит браузеру сжата ли страница и в каком виде
- **Server** и **X-Powered-By** — технологии, на которых работает сайт

HTTP-редиректы

Редиректы — способ перенарпавить браузер на другой URL. Это похоже на то, как если бы вы пришли в магазин и там увидели надпись "мы переехали, вот наш новый адрес". После этого вы идете в магазин по новому адресу. Важно понимать, что редирект со стороны сайта лишь просит ваш браузер перейти на другой адрес, а не подсовывает вам другую страницу. Другими словами, технически редирект происходит на стороне **клиента**, а не **сервера**.

www и http://

Проверка редиректов очень важна для проверки настроек префиксов домена типа `www` и `http`/`https`.

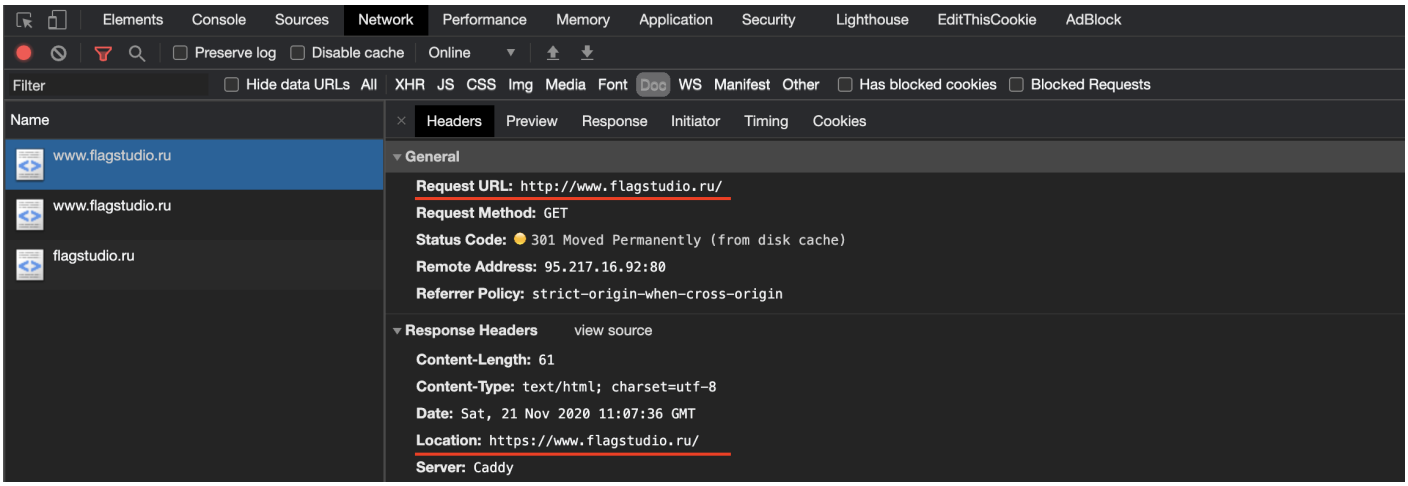
То есть, у каждого домена обычно есть 4 варианта написания:

- http://domain.ru
- http://www.domain.ru
- https://domain.ru (обычно, основной вариант)
- https://www.domain.ru

И 3 из этих 4 вариантов **обязательно** должны ссылаться на один из них, то есть на главный. Главный вариант определяет заказчик, но обычно это https://domain.ru

Инструменты

Редиректы можно проверить браузером через вкладку Network



Разработчикам будет удобнее воспользоваться утилитой

```
curl -I -L https://domain.ru
```

Как сбросить кэш редиректов браузера

На Windows нажать ctrl+H, затем "очистить историю", оставить включенной (отжатой) только галочку напротив "Изображения и другие файлы, сохраненные в кэше". Затем нажать "удалить данные".

SSL шифрование

Принцип — передача только зашифрованных данных.

Типы SSL-сертификатов

- Платные (Comodo, Thawte, reg.ru)
  - Срок — 1 год или более
  - Стоимость — от 1500 руб/год
- Let's Encrypt (обычно, мы используем именно LE)

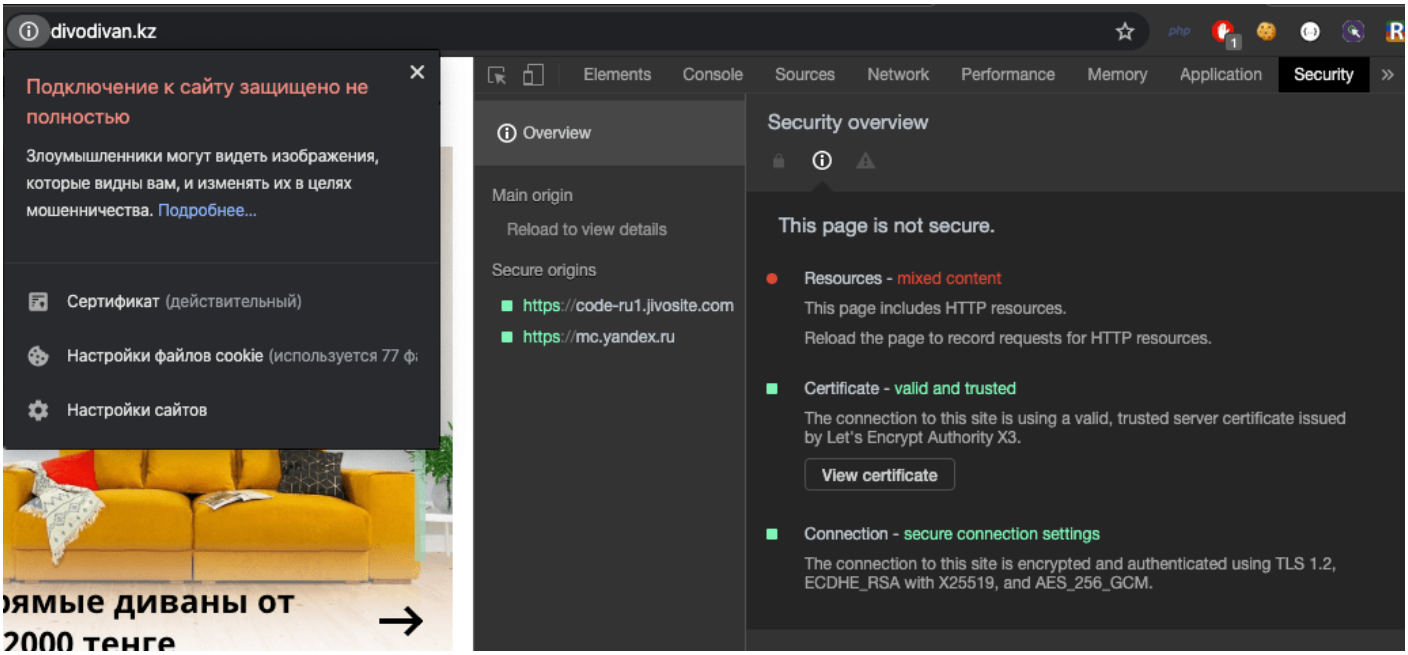
- Срок — 3 месяца
- Стоимость — бесплатно
- Нюанс: Wild Card сертификаты для поддоменов (*.flagsoft.ru). Доступны в LE только при DNS-аутентификации, то есть добавлении TXT-записи, которую нужно обновлять каждые 3 месяца. Для обновления используется API, которое есть **не у всех DNS-провайдеров**.

Диагностика SSL

Если на сайте проблемы с SSL, то это одна из двух проблем

- Что-то не так с сертификатом (кончился, выдан не на этот домен или самоподписанный)
- Страница запрашивает что-то по HTTP

На скриншоте видно, как посмотреть подробности о сертификате и наличие ошибок в странице. Конкретно на этом скриншоте показана вторая проблема, так называемая "mixed-content error", то есть часть контента прилетает по HTTP.



Для диагности SSL на вашем сайте вы также можете использовать онлайн-сервис https://www.ssllabs.com/ssltest/

> {primary} Учтите, данные о сертификате, как и HTTP-редиректы, сильно кэшируются браузером. Сбрасываются также, как кэш редиректов (смотрите выше).

Версии HTTP

HTTP/2

Суть - мультиплексирование запросов по TCP.

Преимущества перед HTTP/1.1

— Скорость. HTTP/2 — бинарный протокол, а HTTP/1.1 — текстовый

- Скорость. Мультиплексирования множества запросов в одном соединении TCP
- Скорость. Сжатие HTTP-заголовков
- Server Push. Принудительная отправка данных сервером в браузер
- Приоритеты запросов

Минусы

- Из-за "head-of-line blocking" на медленном интернете HTTP/1.1 работает быстрее
- Для работы HTTP/2 нужен SSL

Распространение: По данным W3Techs на 1 ноября 2020 года, 49 % из 10 млн самых популярных интернет-сайтов поддерживают протокол HTTP/2.

Подробая статья с понятными картинками https://habr.com/ru/company/nix/blog/304518/

HTTP/3

Суть — мультиплексирование по UDP.

Преимущества перед HTTP/2:

- Решена проблема "head-of-line blocking"
- Шифрование и транспорт объединены, пакеты шифруются независимо
- Протокол реализуется на уровне приложений, а не ОС. Поэтому быстро внедряется

Минусы:

- Протокол реализуется на уровне приложений, а не ОС. Поэтому HTTP/3 использует больше CPU

Результаты: По данным Google, страницы загружаются примерно на 5% быстрее, а в потоковом видео на 30% меньше подвисаний по сравнению с TCP.

Распространение: По данным W3Techs на 1 сентября 2020 года, 7 % из 10 млн самых популярных интернет-сайтов поддерживают протокол HTTP/3.

Вот хорошая статья https://habr.com/ru/company/dododev/blog/473930/

Версия #1
Admin создал 4 December 2022 18:44:21
Admin обновил 19 April 2023 12:35:26