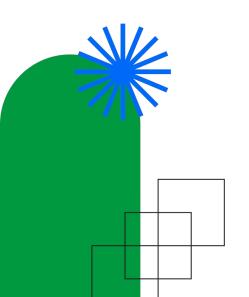
НТТР-протокол









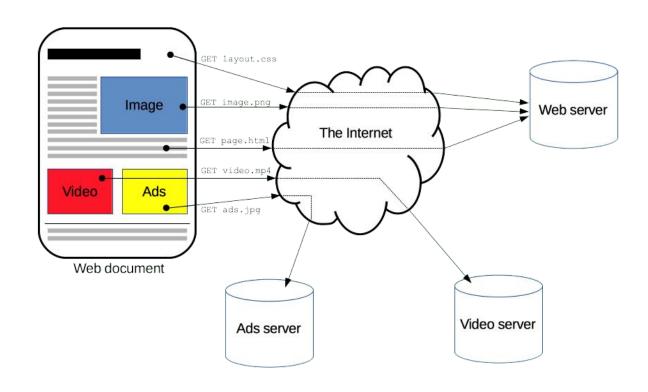
HTTP (HyperText Transfer Protocol)

протокол прикладного уровня передачи данных, изначально — в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных.



НТТР-протокол











- 1. Открытие TCP-соединения: служит для отправки запросов и получения ответов
- 2. Отправка НТТР-сообщения (запроса и ответа)
- 3. Чтение ответа с сервера
- 4. Закрытие ТСР-соединения или отправка следующих запросов в нем же





HTTP-сообщения - обмен данными между сервером и клиентом

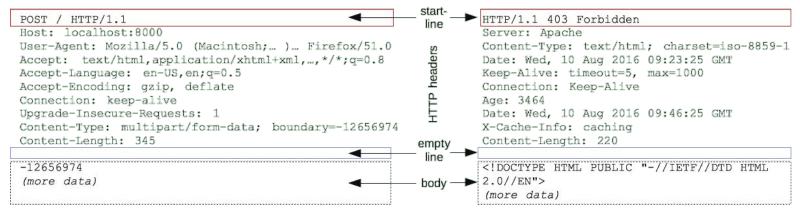
Activity initiation HTTP/2 stream (composed of frames) Frame APIS HTTP/1.x message Type=HEADERS (script) Frame Type=CONTINUATION PUT /create page HTTP/1.1 **UI-activity** Host: localhost:8000 Frame (browser) Connection: keep-alive Translation Upgrade-Insection Content-Type: text/html Binary Type=CONTINUATION Upgrade-Insecure-Requests: 1 framing Frame Content-Length: 345 HTML Forms Type=DATA Body line 1 (browser) Body line 2 Frame Type=DATA Config file Frame (server) Type=DATA





Запросы и ответы

Requests



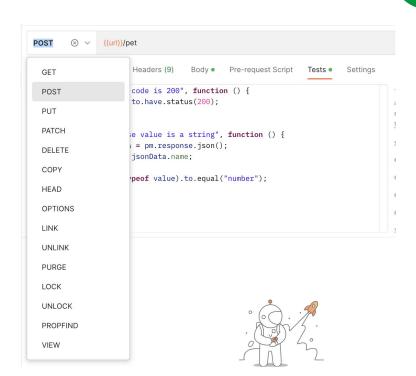




НТТР-методы

НТТР-метод - глагол или существительное, описывающее требуемое действие.

- 1. GET
- 2. POST
- 3. PUT
- 4. PATCH
- 5. DELETE
- 6. OPTIONS
- 7. и т.д.







CRUD-операции

CRUD — акроним, обозначающий четыре базовые функции, используемые при работе с базами данных: создание, чтение, модификация, удаление [Википедия]

C - POST, PUT

R - GET

U - PUT, PATCH

D - DELETE

Также данные операции можно переложить на права пользователя (permission model).





GET vs POST

GET	POST
Получение данных с сайта	Отправка данных на сайт
Отправляется через URL	В теле HTTP-запроса
Тела запроса нет, но его можно добавить. В таком случае оно просто игнорируется	Есть тело запроса
Ограничено длиной адресной строки (до 2048 символов)	Не ограничено (могут быть ограничения сервера)
Данные видны всем (в адресной строке, в истории)	Данные можно увидеть только с помощью программ
Страницу можно добавить в Закладки	Нельзя
Страница может быть кэширована и проиндексирована	Нельзя





PATCH vs PUT

PATCH	PUT
Изменяет ресурс частично	Создает или заменяет ресурс полностью
Неидемпотентный. Ресурс измениться только один раз	Идемпотентный. При отправке нового запроса каждый раз будет создаваться новая сущность





Пример работы различных методов

Создание товара в магазине

- 1. Если отправить 5 одинаковых запросов POST, то создается 5 товаров
- 2. Если отправить 5 одинаковых запросов PUT, то создается только 1 товар
- 3. Если я захочу изменить конкретное название товара, то отправляется РАТСН с запросом на изменение именного этого параметра
- 4. Чтобы найти товар, нужно использовать GET
- 5. Чтобы удалить товар, нужно использовать DELETE





Headers - заголовки

- 1. Служебная информация, дополнительная информация для сервера
- 2. Содержатся как в заголовке, так и в ответе

Несколько примеров:

- :scheme: https
- accept-encoding: gzip, deflate, br
- accept-language: ru-BY,ru;q=0.9,en-BY;q=0.8,en;q=0.7,ru-RU;q=0.6,en-US;q=0.5
- content-type: text/plain
- user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36
- authorization: token
- Set Cookie, cookie





HTTP-body

- 1. В теле передаются данные, полезная нагрузка (payload)
- 2. Request body тело запроса
- 3. Response body тело ответа
- 4. В теле могут передаваться следующие форматы: JSON, XML, form-data, text

```
<
POST
                                                                                                                                   Send
           Authorization
                            Headers (9)
                                              Body •
                                                         Pre-request Script
                                                                                Tests •
                                                                                           Settings
                                                                                                                                        Cookies
Params
          Beautify
                                                                                                 Text
          "id": {{petID}}},
                                                                                                 JavaScript
          "category": {
          · · · "id": · 1.
                                                                                                 JSON
          ···"name": "cats"
                                                                                                 HTML
          "name": " {{$randomFirstName}}",
          "photoUrls": [
                                                                                                 XML
          "https://ru.depositphotos.com/stock-photos/%D0%BA%D0%BE%D1%82.
  10
          ·],
  11
          "tags": [
```





Коды ответа (response status codes)

100	200	300	400	500
Informational	Successful	Redirection	Client errors	Server error
Устарели, запоминать примеры не надо	200 OK 201 Created	301 Moved Permanently 304 Not Modified	400 Bad Request 401 Unauthorized 403 Forbidden 404 Not Found 418 I'm a teapot	500 Internal Server Error 501 Not Implemented 502 Bad Gateway 503 Service Unavailable 504 Gateway Timeout





Версии НТТР-протокола

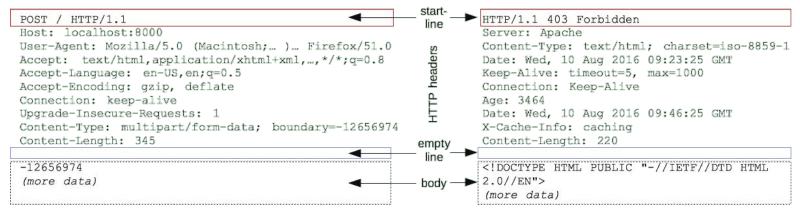
	HTTP 1.1	HTTP 2.0	HTTP 3.0
НТТР-поток	ТСР	ТСР	QUIC
Параллельность отправки запросов	Последовательны е запросы в 1 ТСР	Мультиплексиров ание	Мультиплексиров ание
Headers compression	Нет	Есть	Есть
Приоритет запросов	Нет	Да	Да
Формат	Text	Binary	Binary
Server Push	Нет	Да	Да





Запросы и ответы

Requests







HTTP и HTTPs

- 1. **HTTPs** это расширение для протокола HTTP, которое делает его безопасным
- 2. **HTTPs = HTTP + TLS** (криптографический протокол)
- 3. **SSL-сертификат** цифровое удостоверения сайта

Как и для протокола HTTP у HTTPs есть свои версии:

- 1. HTTPS 1.1 = HTTP + TLS + TCP
- 2. HTTPS 2 = HTTP + TLS + TCP
- 3. HTTPS 3 = HTTP + (TLS + QUIC)

