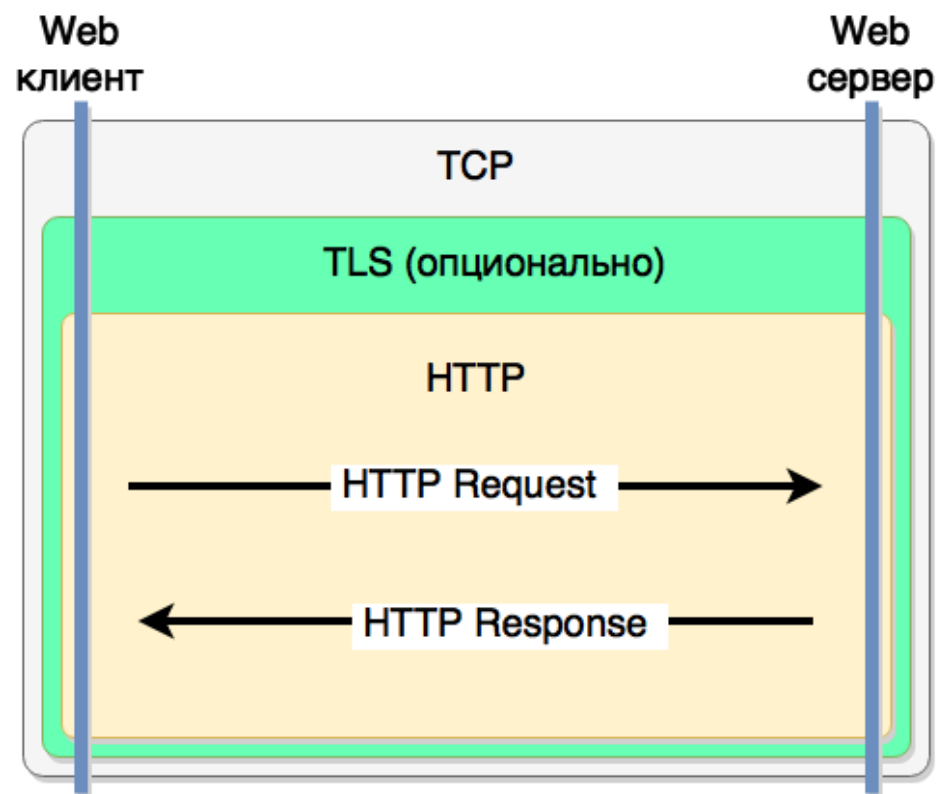


HTTP

Какие задачи решает HTTP?

- Передача документов
- Передача мета-информации
- Авторизация
- Поддержка сессий COOKIE, ПОДДЕРЖКА СОСТОЯНИЯ, СВЯЗАННОГО С КОНКРЕТНЫМ КЛИЕНТОМ
- Кеширование документов (ПРОГРЕВ, ИНВАЛИДАЦИЯ КЕША)
- Согласование содержимого (negotiation)
- Управление соединением



Ключевые особенности HTTP/1.1

- Работает поверх TCP/TLS
- Протокол запрос-ответ
- Не поддерживает состояние (соединение) - **stateless**
- **Текстовый** протокол
- Расширяемый протокол

HTTP/1.0 запрос

GET http://www.ru/robots.txt HTTP/1.0

Accept: text/html, text/plain

User-Agent: curl/7.64.1

If-Modified-Since: Fri, 24 Jul 2015 22:53:05 GMT

МЕТА-ИНФО

НЕ ВИДНА
ПОЛЬЗОВАТЕЛЮ

Перевод строки - \r\n

/R/N

GET - ОТСУТСТВУЕТ ТЕЛО ЗАПРОСА, POST - ЕСТЬ ТЕЛО
ЗАПРОСА

HTTP/1.1 запрос

GET /robots.txt HTTP/1.1

Accept: text/html,application/xhtml+xml

Accept-Encoding: gzip, deflate

Cache-Control: max-age=0

Connection: keep-alive УДЕРЖИВАЕМ СОЕДИНЕНИЕ ПОСЛЕ ПРИХОДА

Host: www.ru ХАРАКТЕРНО ДЛЯ ВЕРСИИ ПРОТОКОЛА
ОТВЕТА

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)

HTTP/1.1 ОТВЕТ

HTTP/1.1 404 Not Found

Server: nginx/1.5.7

Date: Sat, 25 Jul 2015 09:58:17 GMT

Content-Type: text/html; charset=iso-8859-1 MIME-TYPE

Connection: close

/R OR /N

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">


<HTML><HEAD>...

HTTP запрос состоит из

- строка запроса
 - метод
 - URL документа
 - версия
- заголовки
- тело запроса

ПРОТОКОЛОМ HTTP ПРЕДПОЛАГАЕТСЯ, ЧТО GET-ЗАПРОСЫ НЕ МЕНЯЮТ СОСТОЯНИЕ АДРЕСАТА, ПОТОМУ ОНИ МОГУТ КЕШИРОВАТЬСЯ И ПЕРЕОТПРАВЛЯТЬСЯ (НАПРИМЕР, ПРОКСИ-СЕРВЕРОМ) В СЛУЧАЕ НЕУДАЧИ, ТО ЕСТЬ ПРОИЗОЙДЕТ ДУПЛИКАЦИЯ ЗАПРОСА НА ИЗМЕНЕНИЕ СОСТОЯНИЯ --> ДЛЯ ПОДОБНЫХ ВЕЩЕЙ ИСПОЛЬЗУЕМ ТОЛЬКО POST-ЗАПРОС

HTTP методы

- **GET** - получение документа META AND BODY OF THE QUIERY
 - **HEAD** - получение только заголовков (ЗАПРОС ТОЛЬКО МЕТАИНФОРМАЦИИ, БЕЗ ТЕЛА ЗАПРОСА)
 - **POST** - отправка данных на сервер
 - PUT - отправка документа на сервер (*)
 - DELETE - удаление документа (*)
 - CONNECT, TRACE, OPTIONS - используются редко (*)
 - COPY, MOVE, MKCOL - расширения WebDAV (*)
- 
- СЕЙЧАС В ОСНОВНОМ ИСПОЛЬЗУЕТСЯ В REST-FRAMEWORK ПРИ РАБОТЕ С НЕКИМИ СУЩНОСТЯМИ, А НЕ НЕПОСРЕДСТВЕННО С HTML ДОКОМ

И ТД... (МЕТОДЫ
РАСШИРЯЕМЫ)

HTTP ответ состоит из

- строка ответа
 - версия
 - код ответа
 - текстовое описание кода ответа
- заголовки
- тела ответа - документ

HTTP коды ответа

- 1xx - информационные
- 2xx - успешное выполнение
- 3xx - перенаправления
- 4xx - ошибка на стороне клиента
- 5xx - ошибка на стороне сервера

HTTP коды ответа (1)

- 200 OK - запрос успешно выполнен
- 204 No Content - запрос успешно выполнен, но документ пуст
- 301 Moved Permanently - документ сменил URL
- 302 Found - повторить запрос по другому URL
- 304 Not Modified - документ не изменился, использовать кеш

HTTP коды ответа (2)

- 400 Bad Request - неправильный синтаксис запроса
- 401 Unauthorized - требуется авторизация
- 403 Forbidden - нет доступа (неверная авторизация)
- 404 Not Found - документ не найден
- 500 Internal Server Error - неожиданная ошибка сервера (application)
НАРУШЕНА ЛОГИКА РАБОТЫ СЕРВЕРА
- 502 Bad Gateway - проксируемый сервер отвечает с ошибкой
APPLICATION SERVER ДАЖЕ НЕ СЛУШАЕТ НИ НА КАКОМ ПОРТУ
- 504 Gateway Timeout - проксируемый сервер не отвечает

WEB И APPL СЕРВЕР ПОДКЛЮЧЕНЫ, ЗАПРОС
ОТПРАВЛЕН НА APPL SERVER, НО ОТВЕТ НЕ
ПРИХОДИТ В ТЕЧЕНИЕ TIMEOUT'А

Заголовки HTTP (общие)

Для управления соединением и форматом сообщения (документа)

- Content-Type - MIME тип документа
- Content-Length - длина сообщения ТОЛЬКО ДЛИНА ТЕЛА ОТВЕТА/ЗАПРОСА, БЕЗ ДЛИНЫ МЕТАИНФЫ
- Content-Encoding - кодирование документа, например gzip-сжатие
- Transfer-Encoding - формат передачи, например, chunked
- Connection - управление соединением
- Upgrade - смена протокола

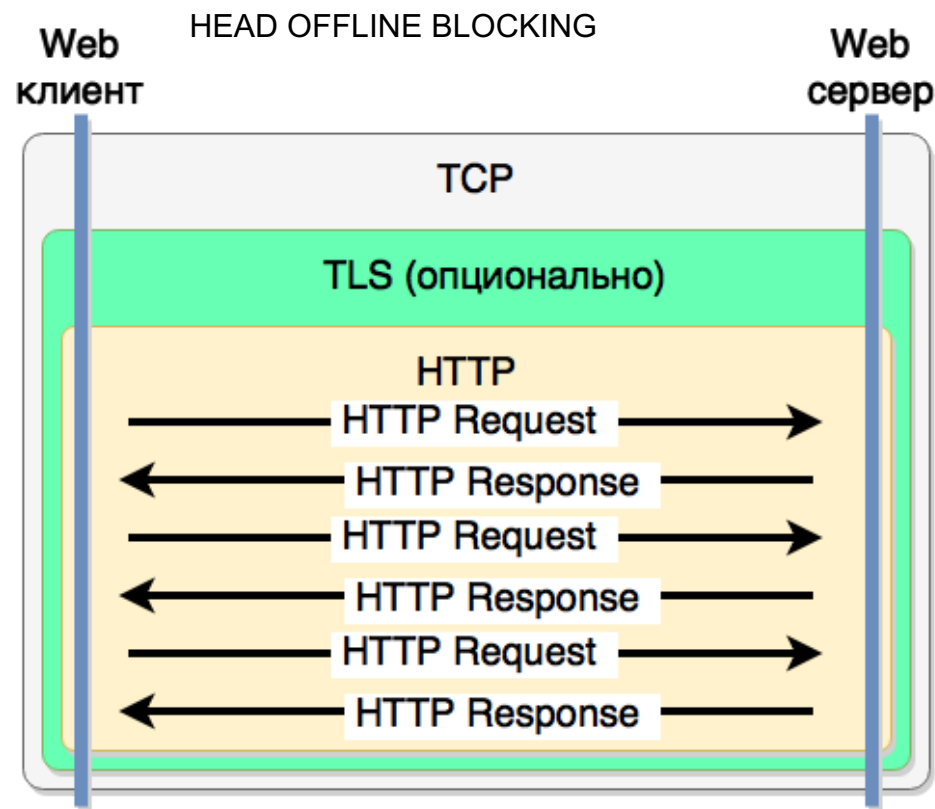
Заголовки HTTP запросов

- Authorization - авторизация, чаще всего логин/пароль
 - Cookie - передача состояния (сессии) на сервер
 - Referer - URL предыдущего документа, контекст запроса
 - User-Agent - описание web-клиента, версия браузера
 - If-Modified-Since - условный GET запрос ЕСЛИ ДОК НЕ ИЗМЕНИЛСЯ В ТЕЧЕНИЕ ОПРЕДЕЛЕННОГО ВРЕМЕНИ -- БЕРЕМ ИЗ КЕША -- КОД 304
 - Accept-* - согласование (negotiation) содержимого
СОГЛАСОВАНИЕ ЯЗЫКА И
ТД
- IF-MODIFIED-SINCE "LAST MODIFIED" -- ЗАПРОС НОВОГО ДОКУМЕНТА(200 ОК), ИНАЧЕ БЕРЕМ ИЗ КЕША(304 NOT MODIFIED)

Заголовки HTTP ответов

- Location - новый URL документа при перенаправлениях В ПАРЕ С 301 И 302: ЗНАЮ, ГДЕ ДОКУМЕНТ, ЭТО ЕГО НОВЫЙ URL
- Set-Cookie - установка состояния (сессии) в браузере
- Last-Modified - дата последнего изменения документа
- Date - Дата на сервере, для согласования кешей
- Server - описание web-сервера, название и версия

HTTP/1.1 управление соединением



НОВОВВЕДЕНИЕ - ОТПРАВКА НЕСКОЛЬКИХ ЗАПРОСОВ И ОТВЕТОВ ЧЕРЕЗ ОДНО ОТКРЫТОЕ СОЕДИНЕНИЕ(НЕ ТРАТИМ ВРЕМЯ НА ПЕРЕОТКРЫТИЕ)

Общие принципы

Протокол HTTP/1.0 предполагает закрытие TCP соединения сразу после ответа сервера.

Протокол HTTP/1.1 предполагает удержание TCP соединения, если не было заголовка `Connection: close`.

Логика управления в HTTP/1.1

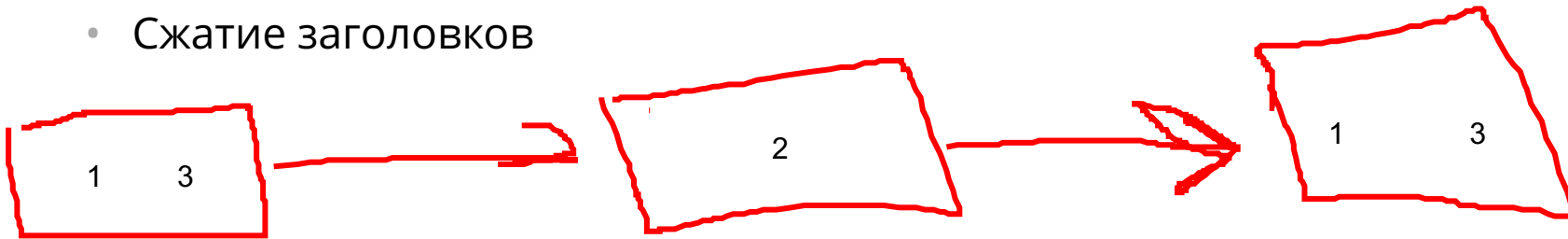
Соединение должно быть закрыто, если:

- сервер или клиент использует HTTP младше 1.1
- сервер или клиент передал заголовок `Connection: close`
- по истечении таймаута (обычно небольшой, около 10с)

Иначе соединение остается открытым для последующих запросов.

Ключевые особенности HTTP/2

- Бинарный протокол уже не текстовый
- Мультиплексирование запросов в одном TCP соединении
- Приоритезация ресурсов
- Сжатие заголовков



ПРОБЛЕМА: Т.К. ПО TCP-СОЕДИНЕНИЮ ПАКЕТЫ ПЕРЕДАЮТСЯ, ПУСТЬ И МУЛЬТИПЛЕКСИРОВАННО, НО ПО-ПРЕЖНЕМУ ПОСЛЕДОВАТЕЛЬНО, НО В СЛУЧАЕ, ЕСЛИ НЕ ДОЙДЕТ ОДИН ИЗ ПАКЕТОВ, ПРОИЗОЙДЕТ ТОТ ЖЕ HEAD OFFLINE BLOCKING: ОСТАЛЬНЫЕ N ПАКЕТОВ БУДУ ЖДАТЬ ПЕРЕОТПРАВКИ -- ПОТОМУ ИСПОЛЬЗУЮТСЯ АЛГОРИТМЫ ОТЛОЖЕННОЙ ОТПРАВКИ НЕДОШЕДШИХ ПАКЕТОВ, РЕАЛИЗОВАННЫЕ В ПРОТОКОЛЕ UDP

Ключевые особенности HTTP/3

- Использует UDP (отказ от TCP)
- Использует новый протокол QUIC ИМЕЕТ ВСЕ ПЛЮСЫ TCP: НАДЕЖНАЯ ДОСТАВКА, ДУПЛЕКСИРОВАНИЕ...
- Мультиплексирование запросов на транспортном уровне

UDP + QUIC В СВЯЗКЕ

