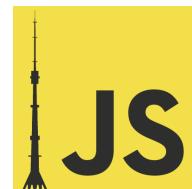


Moscow JS Conf

HTTP/2

Что нужно знать фронтендеру





Александр Майоров

Developer advocate & technical evangelist in
Tutu.ru

- twitter, github: [@frontdevops](https://github.com/@frontdevops)
- alexander@majorov.su





Самый посещаемый сервис туристических услуг в России
(по результатам исследования comScore).

Мы продаем туры, ж/д и авиабилеты, билеты на автобусы,
бронируем отели, рассказываем о расписании.



Авиабилеты



Ж/д билеты



Электрички



Туры



Гостиницы



Автобусы

600 тыс.

посетителей в день

2003 г.

год основания

11 млн

посетителей в месяц

267

сотрудников

HTTP/2



Факты про HTTP/2

- Основан на базе Google SPDY
- Стандартизован в мае 2015 (RFC 7540, 7541)
- Может работать поверх TCP (h2c), но браузеры умеют только поверх TLS (h2 with ALPN)
- Больше не будет минорных версий



TLS или SSL ?

TLS - transport layer security, основан на протоколе SSL (Secure Sockets Layer).

На данный момент последняя версия – TLS 1.2, выпущена в августе 2008 года.

Простым языком: TLS это другая версия SSL и он лучше :)



Ключевые особенности HTTP/2

- Транспортная надстройка над HTTP/1.x
- Сжатие заголовков
- Бинарный формат
- Мультиплексирование
- Приоритезация



Надстройка над HTTP/1.x

▼ General

Request URL: <https://sapsan.tutu.ru/>
Request Method: GET
Status Code: 200
Remote Address: 10.3.3.125:443

▼ Response Headers

content-encoding: gzip
content-length: 18391
content-security-policy-report-only: frame-ancestors 'none'; report-uri /csp_logger/;
content-type: text/html; charset=utf-8
date: Thu, 22 Sep 2016 09:40:33 GMT
p3p: CP="NOI ADM DEV COM NAV OUR STP"
server: nginx
status: 200
vary: Accept-Encoding

▼ Request Headers

:authority: sapsan.tutu.ru
:method: GET
:path: /
:scheme: https
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
accept-encoding: gzip, deflate, sdch, br
accept-language: ru-RU,ru;q=0.8,en-US;q=0.6,en;q=0.4
cache-control: no-cache

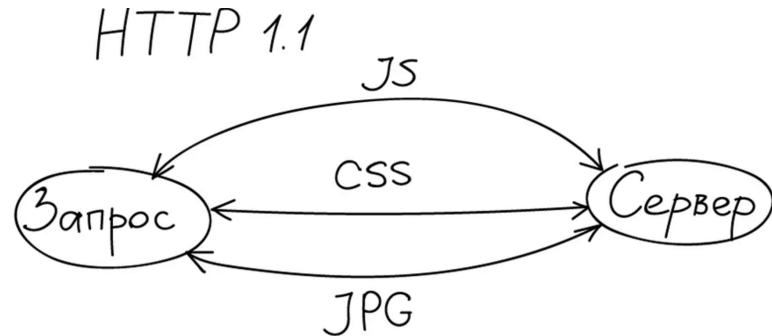
Бинарный формат

```
GET / HTTP/1.1
Host: sapsan.tutu.ru
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/5.0 AppleWebKit/537 Chrome/116 Safari/36
Accept: text/html,application/xhtml+xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: ru,en-US;q=0.6,zh;q=0.4,zh-TW;q=0.2,zh-CN;q=0.2
```

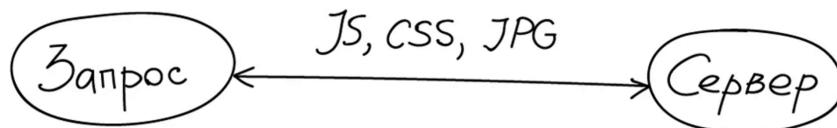
HTTP/2

```
00 00 93 01 25 00 00 00 01 80 00 00 00 B6 82 41 ..% .. .A
8A 90 B4 9D 7A A6 35 5E 57 21 E9 87 00 84 B9 58 ....z.5^W!... .X
D3 3F 85 61 09 1A 6D 47 7A D0 D0 7F 66 A2 81 B0 .?.a..mGz...f...
DA E0 53 FA FC 08 7E D4 CE 6A AD F2 A7 97 9C 89 ..S...~..j.....
C6 BF B5 21 AE BA 0B C8 B1 E6 32 58 6D 97 57 65 ...!.....2Xm.We
C5 3F AC D8 F7 E8 CF F4 A5 06 EA 55 31 14 9D 4F .?.....U1..O
FD A9 7A 7B 0F 49 58 6C 0B 81 71 38 E0 57 71 C5 ..z{.IX1..q8.Wq.
37 0E 51 D8 66 1B 65 D5 D9 73 53 03 2A 2F 2A 50 7.Q.f.e..ss.*/*P
8E 9B D9 AB FA 52 42 CB 40 D2 5F A5 11 21 27 51 ....RB.@._..!'Q
8B 2D 4B 70 DD F4 5A BE FB 40 05 DE -.Kp..Z..@..
```

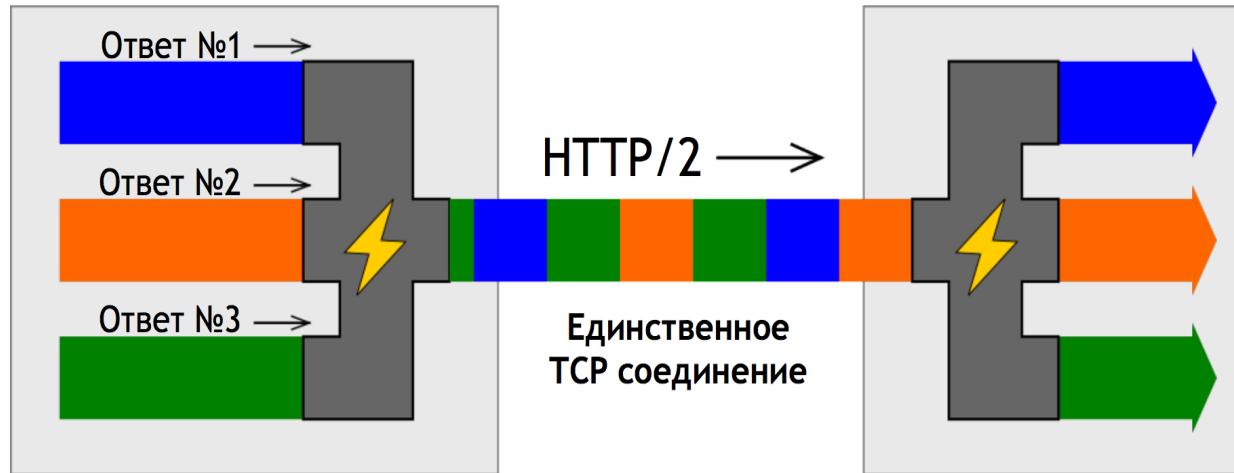
Мультиплексирование



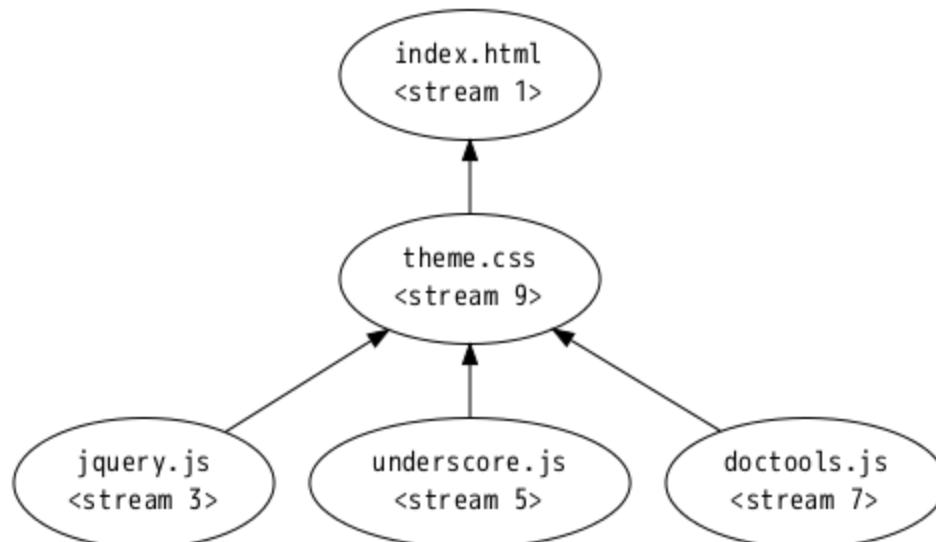
HTTP/2



Мультиплексирование



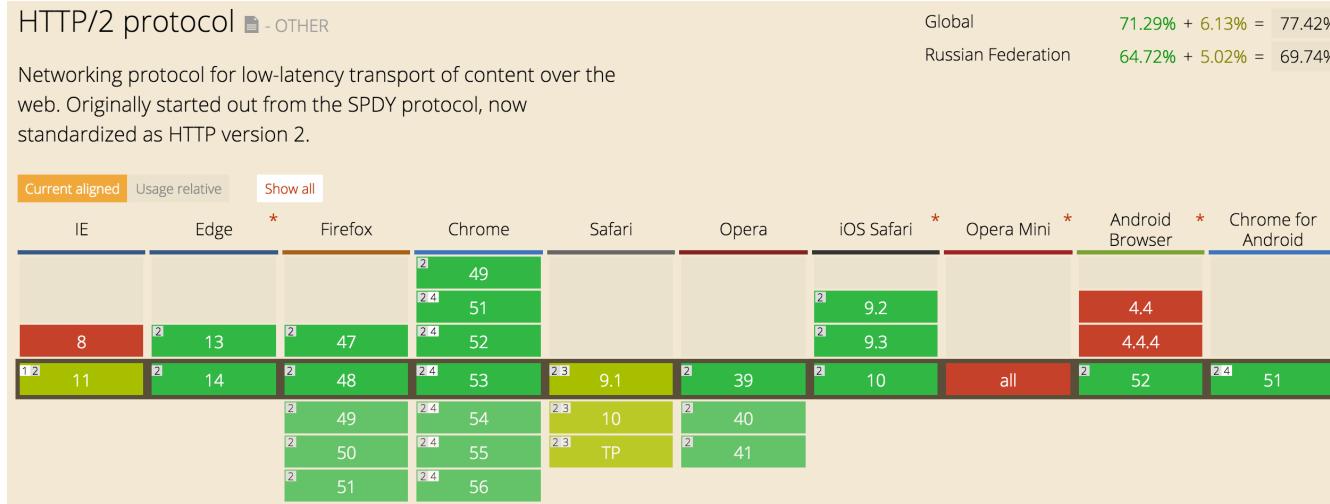
Приоритетизация



SUPPORT

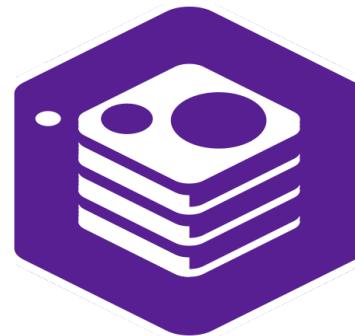
Что с поддержкой?

Поддержка браузерами



Поддержка на бекенде

- Nginx (1.9.5+)
- Apache (2.4.17+) with mod_http2
- Nodejs with http2 module
- **Nghttpx**
- H2O



СЕРТИФИКАТЫ



HTTPS://

Где брать сертификаты ?

- Купить...
- Получить бесплатно у
 - Let's encrypt
 - StartSSL
- Создать самому и подписать (self-signed)



Self-signed сертификаты

- Работает, но **Chrome** запускать с флагом:
chrome --ignore-certificate-errors



SERVER HINTS
RESOURCE HINTS
SERVER PUSH▶

Что это?

```
<link rel="dns-prefetch" href="//example.com">
<link rel="preconnect" href="//example.com">
<link rel="prefetch" href="/next-page.html" as="html">
<link rel="prefetch" href="/library.js" as="script">
<link rel="prefetch" href="/theme.css" as="style">
<link rel="subresource" href="/page.css">
```

Resource hints

```
<link rel="dns-prefetch" href="//example.com">
<link rel="preconnect" href="//example.com">
<link rel="prefetch" href="/next-page.html" as="html">
<link rel="prefetch" href="/library.js" as="script">
<link rel="prefetch" href="/theme.css" as="style">
<link rel="subresource" href="/page.css">
```

Server hints

GET HTTP/1.1

...

Link: <https:// example.com>; rel=dns-prefetch

Link: <https://example.com>; rel=preconnect

Link: <https://example.com/next-page.html>; rel=prerender;

Link: <https://example.com/logo.jpg>; rel=prefetch; as=image;

...

Preload

consumer

<audio>, <video>
<script>, Worker's *importScripts*
<link rel=stylesheet>, CSS @import
CSS @font-face
, <picture>, srcset, imageset
SVG's <image>, CSS *-image
XHR, fetch
Worker, SharedWorker
<embed>
<object>
<iframe>, <frame>

Preload directive

<link rel=preload as=media href=...>
<link rel=preload as=script href=...>
<link rel=preload as=style href=...>
<link rel=preload as=font href=...>
<link rel=preload as=image href=...>
<link rel=preload as=image href=...>
<link rel=preload href=...>
<link rel=preload as=worker href=...>
<link rel=preload as=embed href=...>
<link rel=preload as=object href=...>
<link rel=preload as=document href=...>

Preload

```
<link rel="preload" href="/next-page.html" as="html">
<link rel="preload" href="/library.js" as="script">
<link rel="preload" href="/theme.css" as="style">
<link rel="preload" href="/page.css" as="style">
```

Preload

GET HTTP/1.1

...

Link: </next-page.html>; rel=preload; as=html;

Link: </logo.jpg>; rel=preload; as=image;

Link: </app/script.js>; rel=preload; as=script;

Link: </app/style.css>; rel=preload; as=style;

...

Server push use preload

GET HTTP/1.1

...

Link: </next-page.html>; rel=preload; as=html;

Link: </logo.jpg>; rel=preload; as=image;

Link: </app/script.js>; rel=preload; as=script;

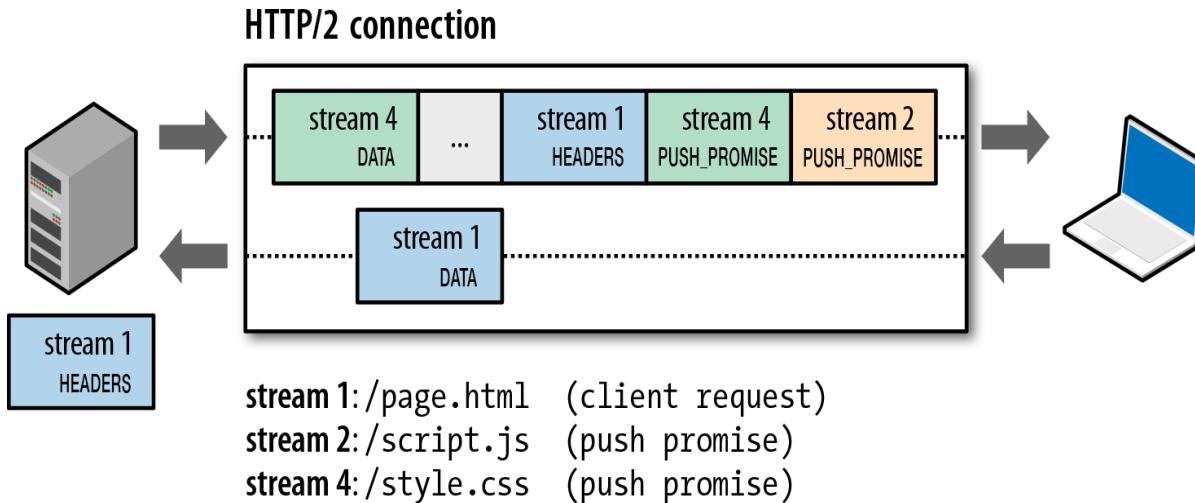
Link: </app/style.css>; rel=preload; as=style; **nopush**

...

RFC5988

Работает в прокси Nghttpx

HTTP/2 Server push

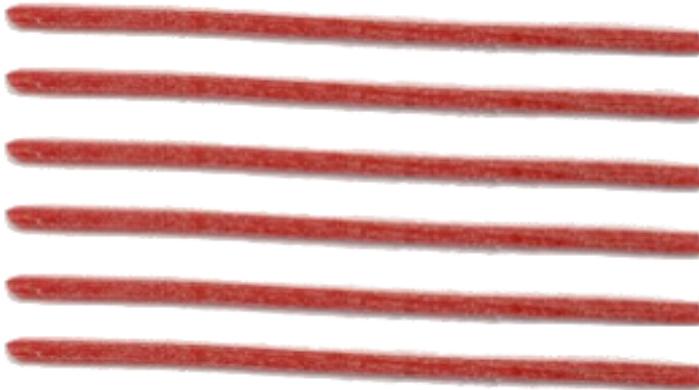


HTTP/2 vs HTTP/1

HTTP/2



HTTP 1.1



Server Push in Application

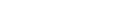
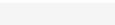
```
const fs    = require('fs'),
      http = require('http2');

var server = http.createServer(options, (request, response)=>{
  let push = response.push('/push.css',
                           {'content-type': 'text/css'});

  fs.createReadStream(`style.css`).pipe(push);
  // или
  push.end(fs.readFileSync(`style.css`));
}

server.listen(...);
```

HTTP/1.1 Secure

| Name | Status | Protocol | Type | Initiator | Size | Time | Timeline – Start Time |
|-----------------|--------|----------|------------|------------------|-------|------|---|
| test.tutu.ru | 200 | http/1.1 | document | Other | 1.0KB | 77ms |  |
| style.css | 200 | http/1.1 | stylesheet | <u>(index):6</u> | 727B | 16ms |  |
| prefetch.css | 200 | http/1.1 | stylesheet | <u>(index):7</u> | 224B | 31ms |  |
| subresource.css | 200 | http/1.1 | stylesheet | <u>(index):8</u> | 227B | 48ms |  |
| preload.css | 200 | http/1.1 | stylesheet | <u>(index):9</u> | 224B | 39ms |  |
| favicon.ico | 200 | http/1.1 | text/plain | Other | 139B | 12ms |  |

Без Server push

| Name | Status | Protocol | Type | Initiator | Size | Time | Timeline – Start Time |
|-----------------|--------|----------|------------|-----------|-------|-------|---|
| test.tutu.ru | 200 | h2 | document | Other | 961 B | 87 ms |  |
| style.css | 200 | h2 | stylesheet | (index):6 | 620 B | 26 ms |  |
| prefetch.css | 200 | h2 | stylesheet | (index):7 | 111 B | 26 ms |  |
| subresource.css | 200 | h2 | stylesheet | (index):8 | 114 B | 27 ms |  |
| preload.css | 200 | h2 | stylesheet | (index):9 | 111 B | 27 ms |  |
| favicon.ico | 200 | h2 | text/plain | Other | 61 B | 32 ms |  |

Server push в действии

| Name | Status | Protocol | Type | Initiator | Size | Time | Timeline – Start Time |
|-----------------|--------|----------|------------|------------------|--------|--------|---|
| test.tutu.ru | 200 | h2 | document | Other | 1.1 KB | 103 ms |  |
| style.css | 200 | h2 | stylesheet | Push / (index):6 | 612 B | 2 ms |  |
| prefetch.css | 200 | h2 | stylesheet | Push / (index):7 | 110 B | 3 ms |  |
| subresource.css | 200 | h2 | stylesheet | Push / (index):8 | 113 B | 3 ms |  |
| preload.css | 200 | h2 | stylesheet | Push / (index):9 | 110 B | 4 ms |  |
| favicon.ico | 200 | h2 | text/plain | Other | 38 B | 31 ms |  |

ОТЛАДКА

Инструменты отладки HTTP/2



Инструменты отладки HTTP/2

- nghttp
- curl
- openssl
- h2c
- h2i

Инструменты отладки OpenSSL

```
openssl s_client -connect sapsan.tutu.ru:443 -nextprotoneg "
```

```
CONNECTED(00000003)
```

```
Protocols advertised by server: h2, http/1.1
```

```
[ ... ]
```

Инструменты отладки CURL

```
curl -vso /dev/null --http2 https://sapsan.tutu.ru
```

```
[ ... ]
* Using HTTP2, server supports multi-use
* Connection state changed (HTTP/2 confirmed)
* TCP_NODELAY set* Copying HTTP/2 data in stream ...
[ ... ]
```

Инструменты отладки h2c

- h2c start --dump
- h2c connect sapsan.tutu.ru
- h2 get /
- h2 disconnect

Вывод информации в h2c

```
[ 09:24 <mayorov@mruapps> ~ ]
[ bash: h2c start --dump &
[1] 14430

[ 09:24 <mayorov@mruapps> ~ ]
[ bash: h2c connect sapsan.tutu.ru
-> SETTINGS(0)
  - ACK
  {empty}

<- SETTINGS(0)
  - ACK
  SETTINGS_MAX_CONCURRENT_STREAMS: 128
  SETTINGS_INITIAL_WINDOW_SIZE: 65536
  SETTINGS_MAX_FRAME_SIZE: 16777215

<- WINDOW_UPDATE(0)
  Window size increment: 2147418112

-> SETTINGS(0)
  + ACK
  {empty}

<- SETTINGS(0)
  + ACK
  {empty}

[ 09:25 <mayorov@mruapps> ~ ]
[ bash: h2c get /
```

nghttp2

nghhttp -nv https://sapsan.tutu.ru

```
[ bash: nghhttp -nv https://sapsan.tutu.ru
[ 0.051] Connected
The negotiated protocol: h2
[ 0.146] recv SETTINGS frame <length=18, flags=0x00, stream_id=0>
(niv=3)
[SETTINGS_MAX_CONCURRENT_STREAMS(0x03):128]
[SETTINGS_INITIAL_WINDOW_SIZE(0x04):65536]
[SETTINGS_MAX_FRAME_SIZE(0x05):16777215]
[ 0.146] recv WINDOW_UPDATE frame <length=4, flags=0x00, stream_id=0>
>window_size_increment=2147419112
(dep_stream_id=7, weight=1, exclusive=0)
[ 0.146] send PRIORITY frame <length=5, flags=0x00, stream_id=11>
(dep_stream_id=3, weight=1, exclusive=0)
[ 0.146] send HEADERS frame <length=38, flags=0x25, stream_id=13>
; END_STREAM | END_HEADERS | PRIORITY
(padlen=0, dep_stream_id=11, weight=16, exclusive=0)
; Open new stream
:method: GET
:path: /
:scheme: https
:authority: sapsan.tutu.ru
:accept: */*
:accept-encoding: gzip, deflate
:user-agent: nghhttp2/1.14.1
[ 0.192] recv SETTINGS frame <length=0, flags=0x01, stream_id=0>
; ACK
(niv=0)
[ 1.049] recv (stream_id=13) :status: 200
[ 1.049] recv (stream_id=13) server: nginx
[ 1.049] recv (stream_id=13) date: Thu, 22 Sep 2016 06:34:38 GMT
[ 1.049] recv (stream_id=13) content-type: text/html; charset=utf-8
[ 1.049]
```

HTTP/2

Нужен ли?

новый модный трололол 😊

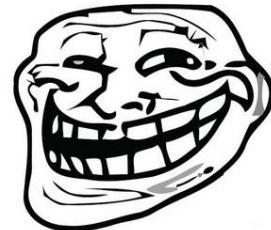
Трудности перехода на HTTP/2

- Высокий порог входа по сравнению с HTTP/1.x
- Необходимость сертификатов
- Затраты на шифрование
- Текущие оптимизации становятся Bad practice
- Не все браузеры поддерживают
- Не всё серверное ПО реализовало весь стандарт целиком
- Server Push – пуля, которая может убить ваш сервис

Нужен ли HTTP/2 ?

Бизнес ответит:

Нужен **HTTPS**



Зачем переходить на HTTPS

- Новые браузерные фичи (Service Workers)
- SEO
- Аналитика
- Повышение конверсии
- Что-то еще, что я забыл или сам еще не знаю...

Зачем переходить на HTTPS

- Новые браузерные фичи (Service Workers)
- SEO
- Аналитика
- Повышение конверсии
- Что-то еще, что я забыл или сам еще не знаю...

А HTTP/2 нужен?



HTTP/2 нужен

- Снижает оверхед от использования шифрования
- Ниже загрузка ЦПУ и расход памяти
- Можно использовать конвейерную обработку запросов и ответов
- Снижает вероятность перегрузки сети (меньше TCP соединений)
- Уменьшает лаги для последующих запросов (не нужно заново устанавливать TCP соединение).
- Отпадает необходимость в доменном шардировании
- Ускорение загрузки страниц



**Спасибо!
Вопросы?**



Ссылки по теме и на слайды

<https://goo.gl/uqdQeC>





Александр Майоров

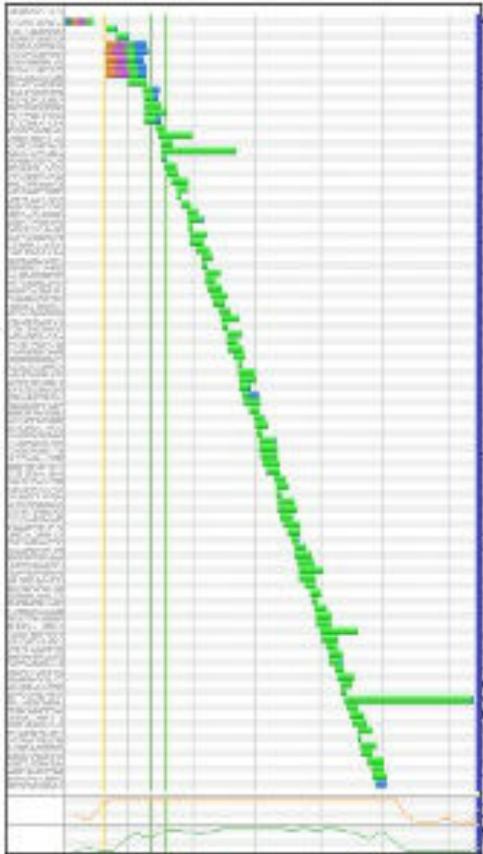
Developer advocate & technical evangelist in
Tutu.ru

- twitter, github: [@frontdevops](https://github.com/@frontdevops)
- alexander@majorov.su

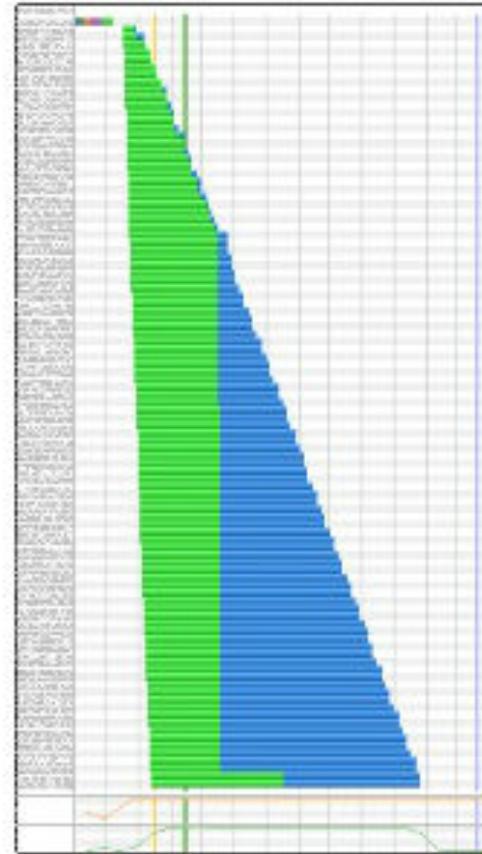


Секретные слайды

HTTP/1.1



HTTP/2



VS