WebTransport и его место среди других протоколов

Андрей Власов

tg: @InnoAndrez
E-mail: innoavvlasov@gmail.com

Василий Маркитан

tg: @Markitan_V
E-mail: mkt.dev@yandex.ru

SBER

15 + стаж разработки

Лидер компетенций WEB

Региональный директор разработки СБЕР в респ. Татарстан

Преподаю в университете Иннополис

Андрей Власов

tg: @InnoAndrez
E-mail: innoavvlasov@gmail.com

7 + стаж разработки

Руководитель направления разработки JS

5 + стаж преподавания

Василий Маркитан

tg: @Markitan_V
E-mail: mkt.dev@yandex.ru

SBER

Современный Web

Онлайн конференции

Оркестровка онлайн игр

Облачный gaming

Broadcasting

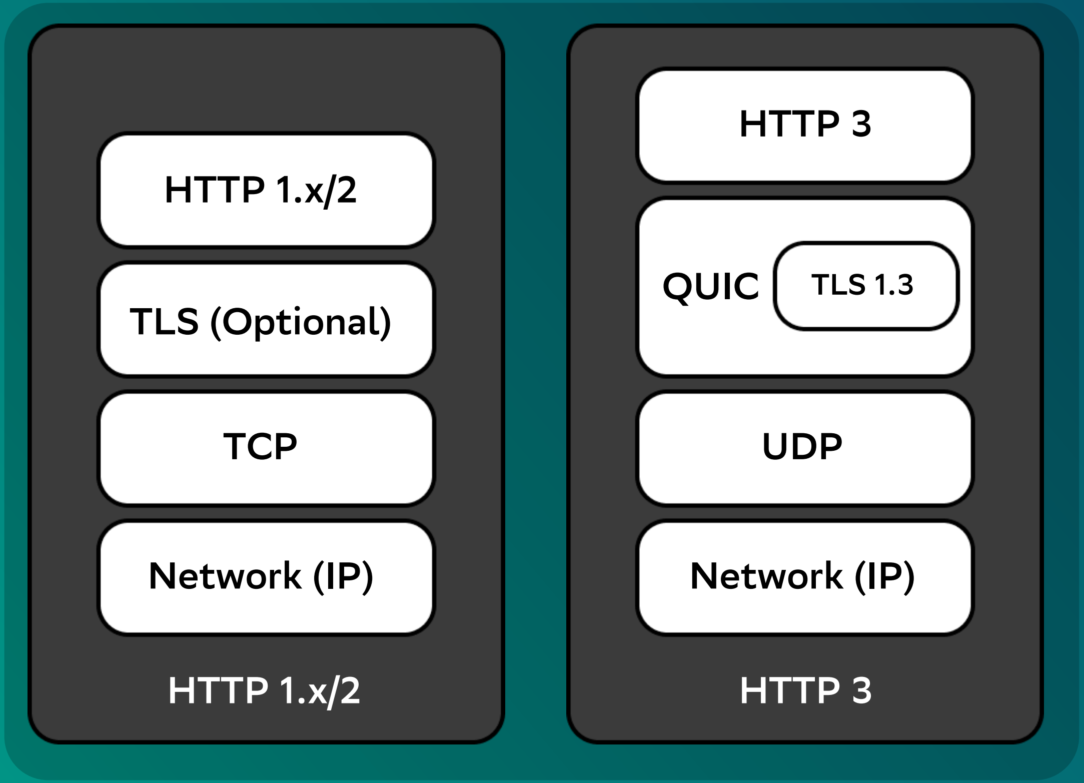
IOT датчики

Joe Carlson
Sí, los leí anoche. Todo parecía bastante claro, pero tenía una o dos preguntas.

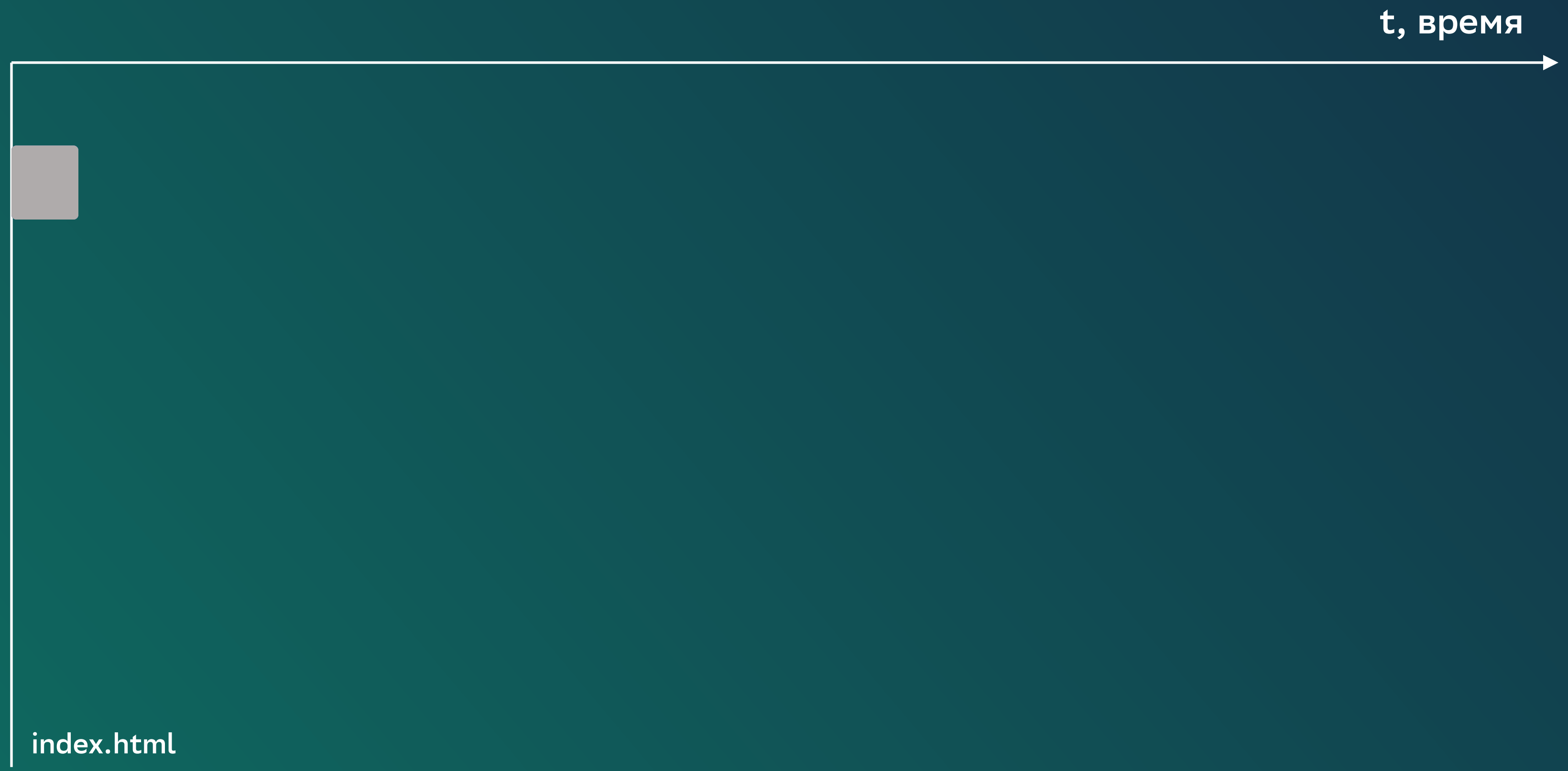
Перевод речи

Протоколы

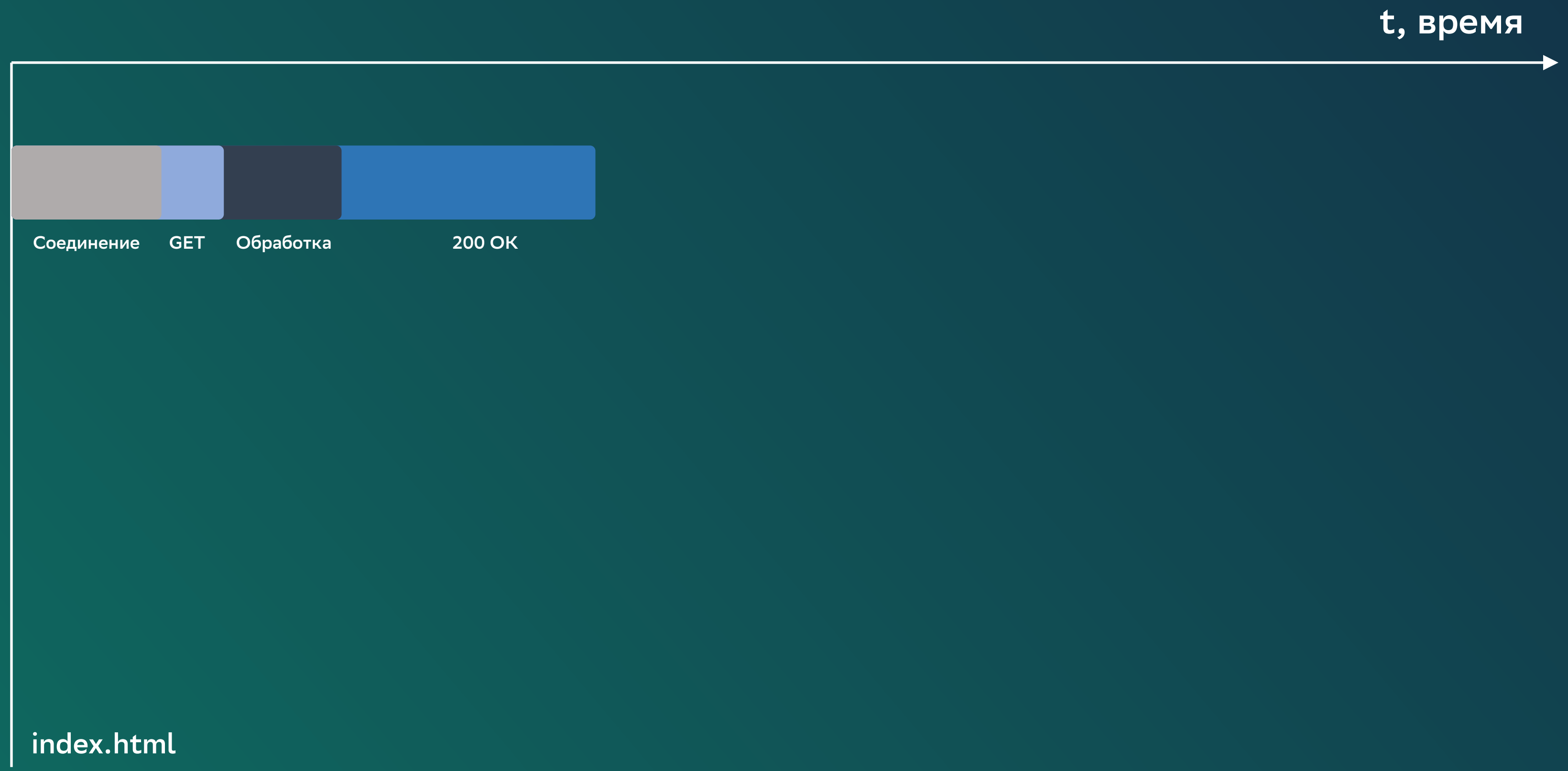
HTTP


HTTP 1.x/2
TLS (Optional)
TCP
Network (IP)
HTTP 1.x/2
HTTP 3
QUIC
TLS 1.3
UDP
Network (IP)
HTTP 3

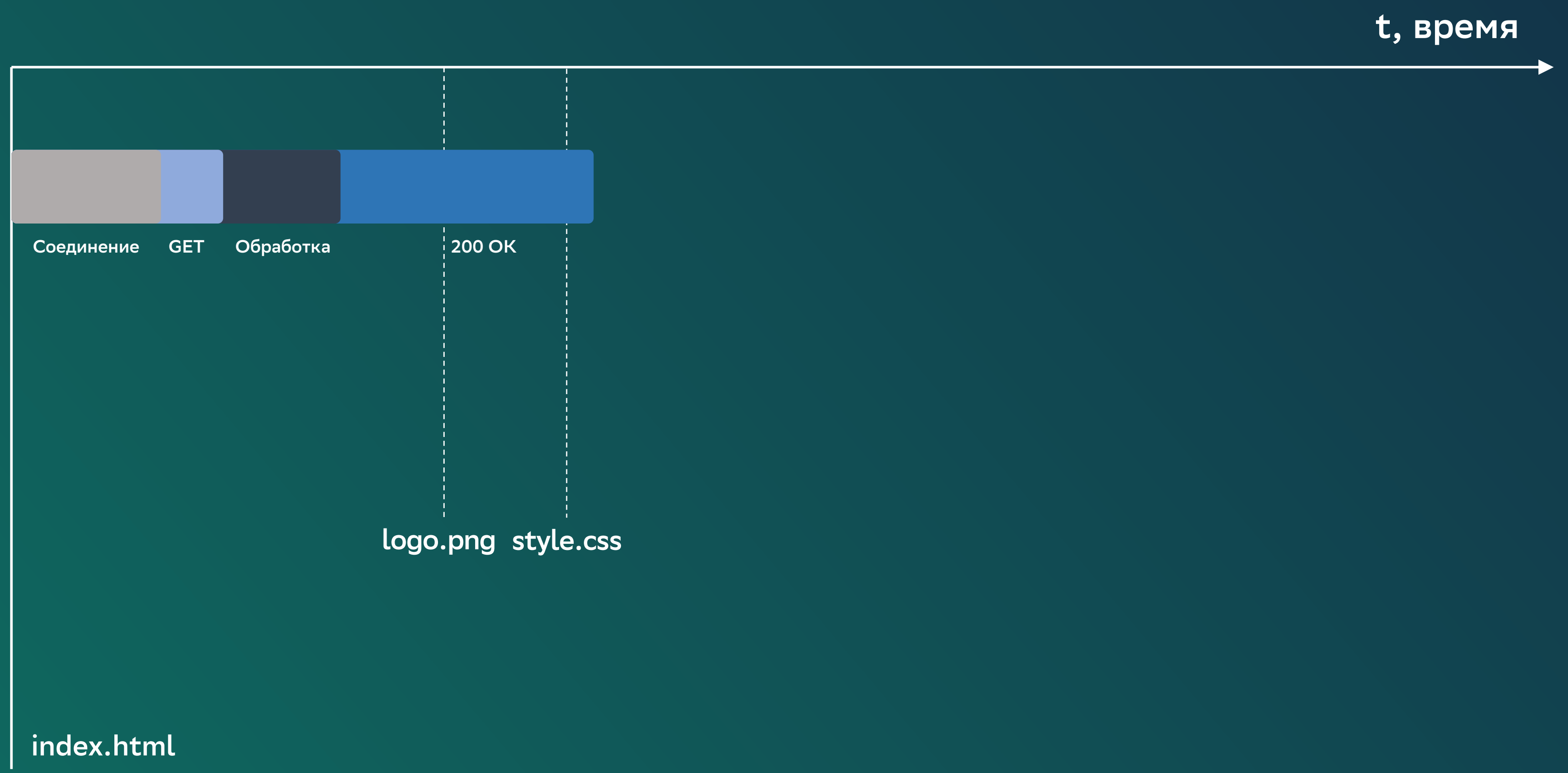
HTTP



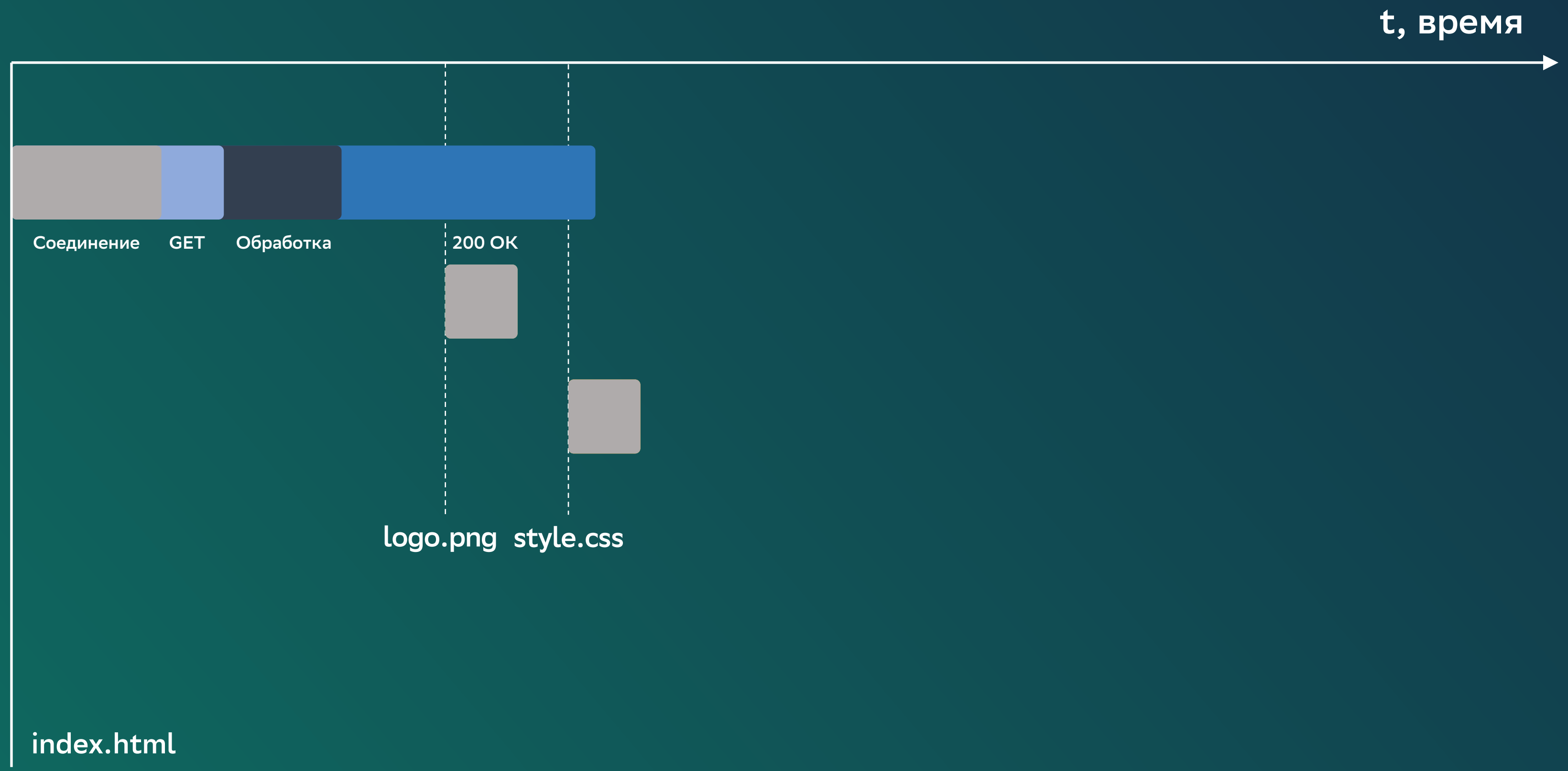
HTTP



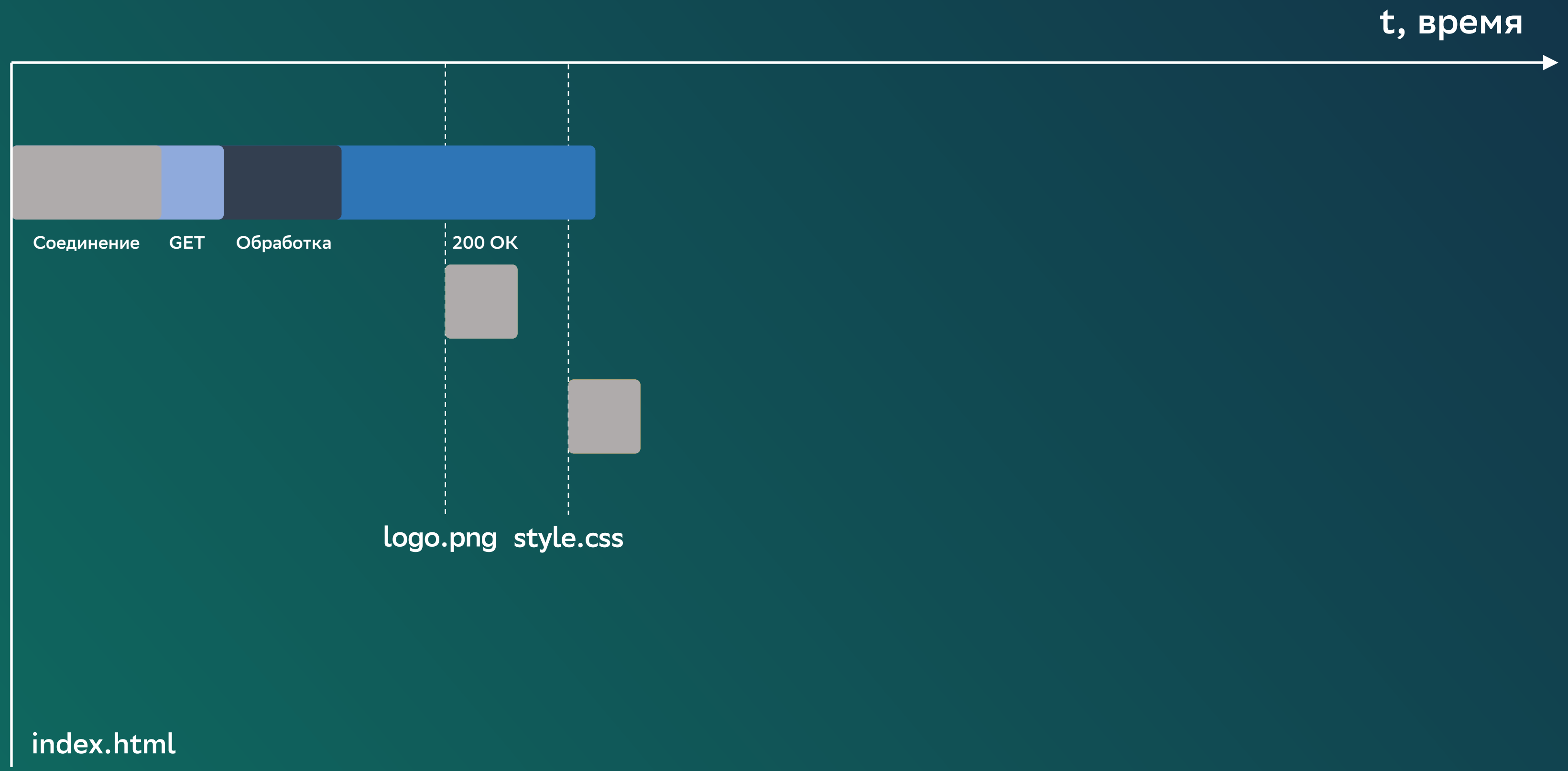
HTTP


t, время
Соединение
GET
Обработка
200 OK
logo.png
style.css
index.html

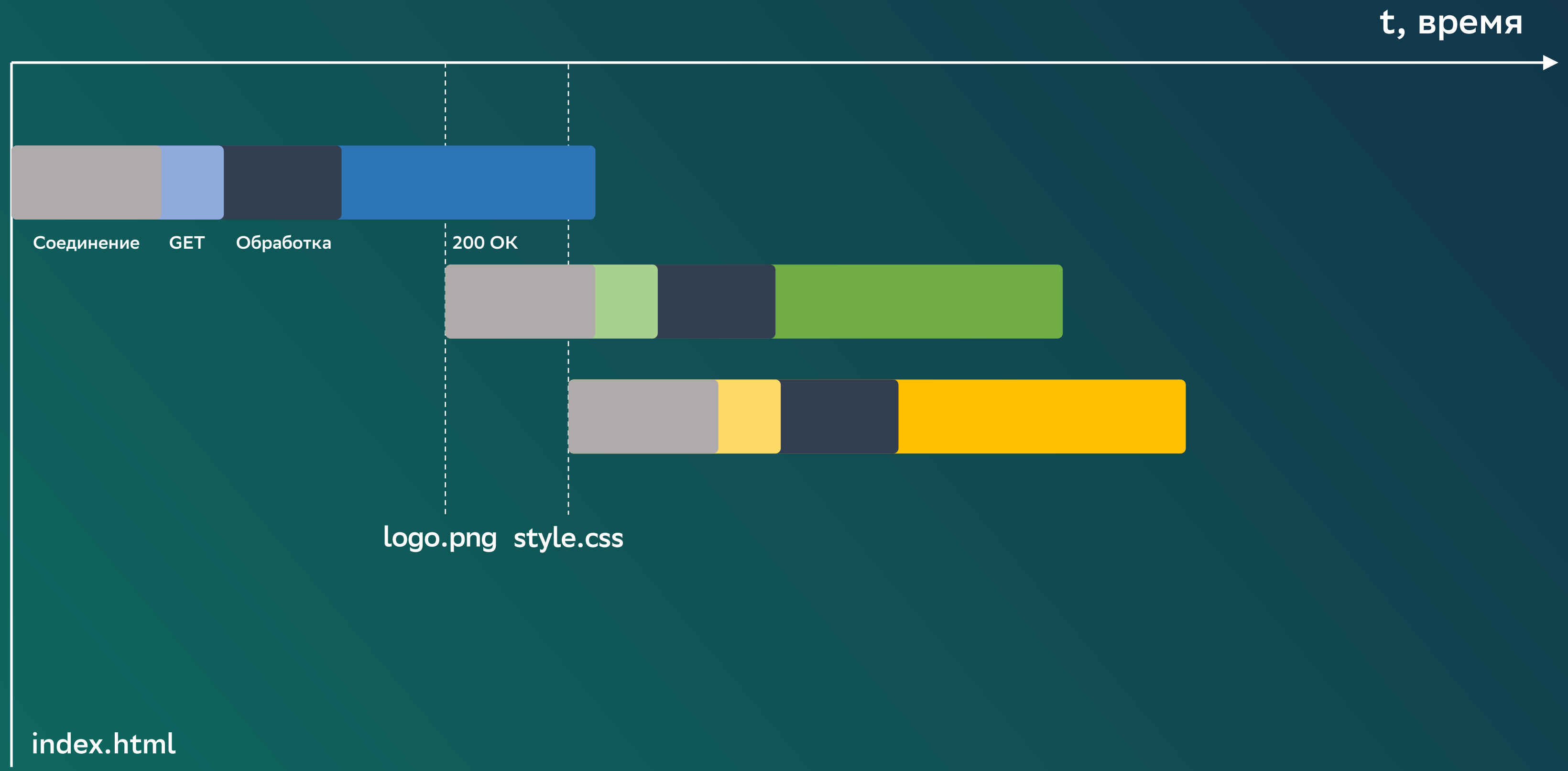
HTTP


t, время
Соединение
GET
Обработка
200 OK
logo.png
style.css
index.html

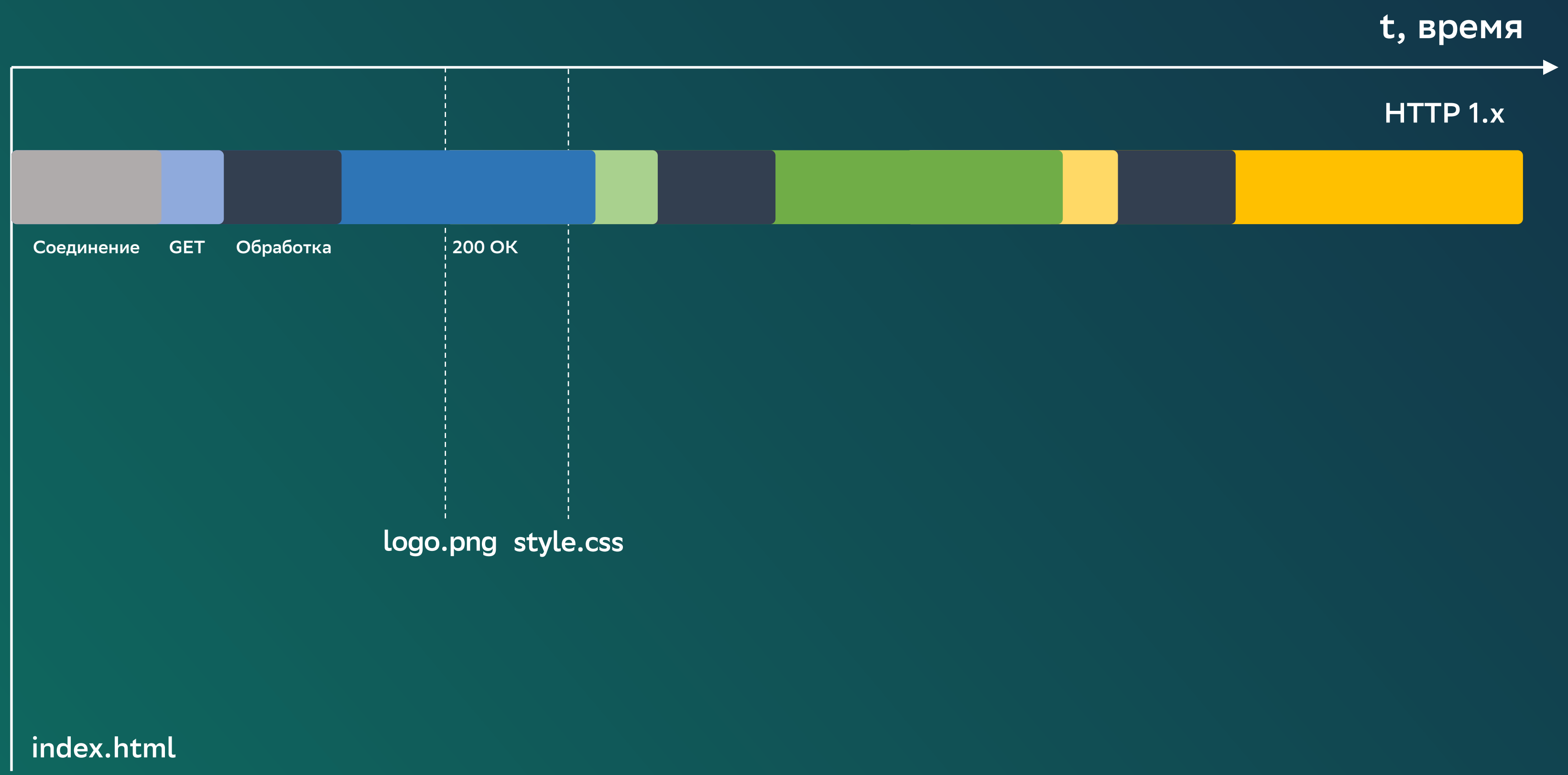
HTTP


t, время
Соединение
GET
Обработка
200 OK
logo.png
style.css
index.html

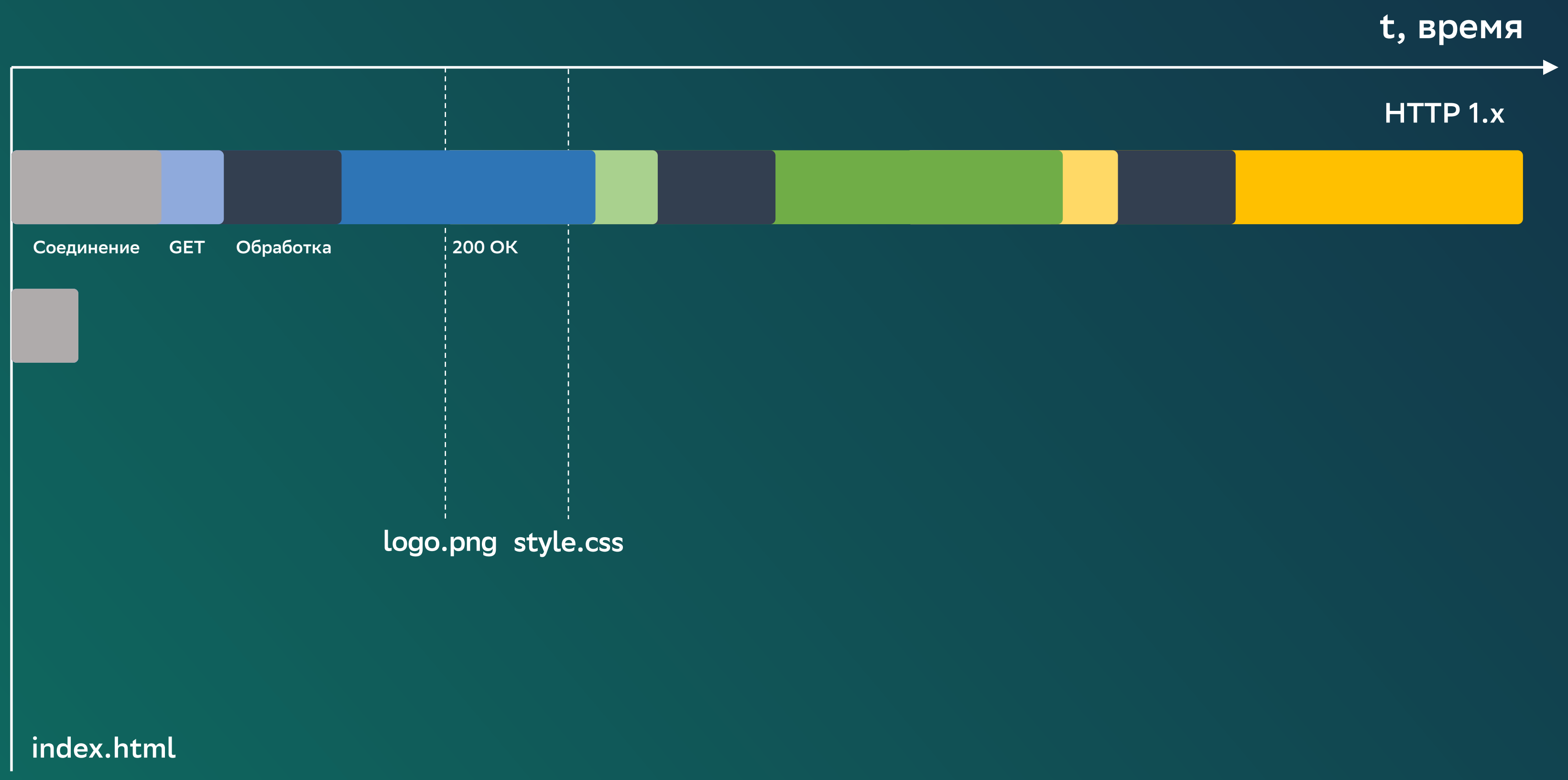
HTTP


t, время
Соединение
GET
Обработка
200 OK
logo.png
style.css
index.html

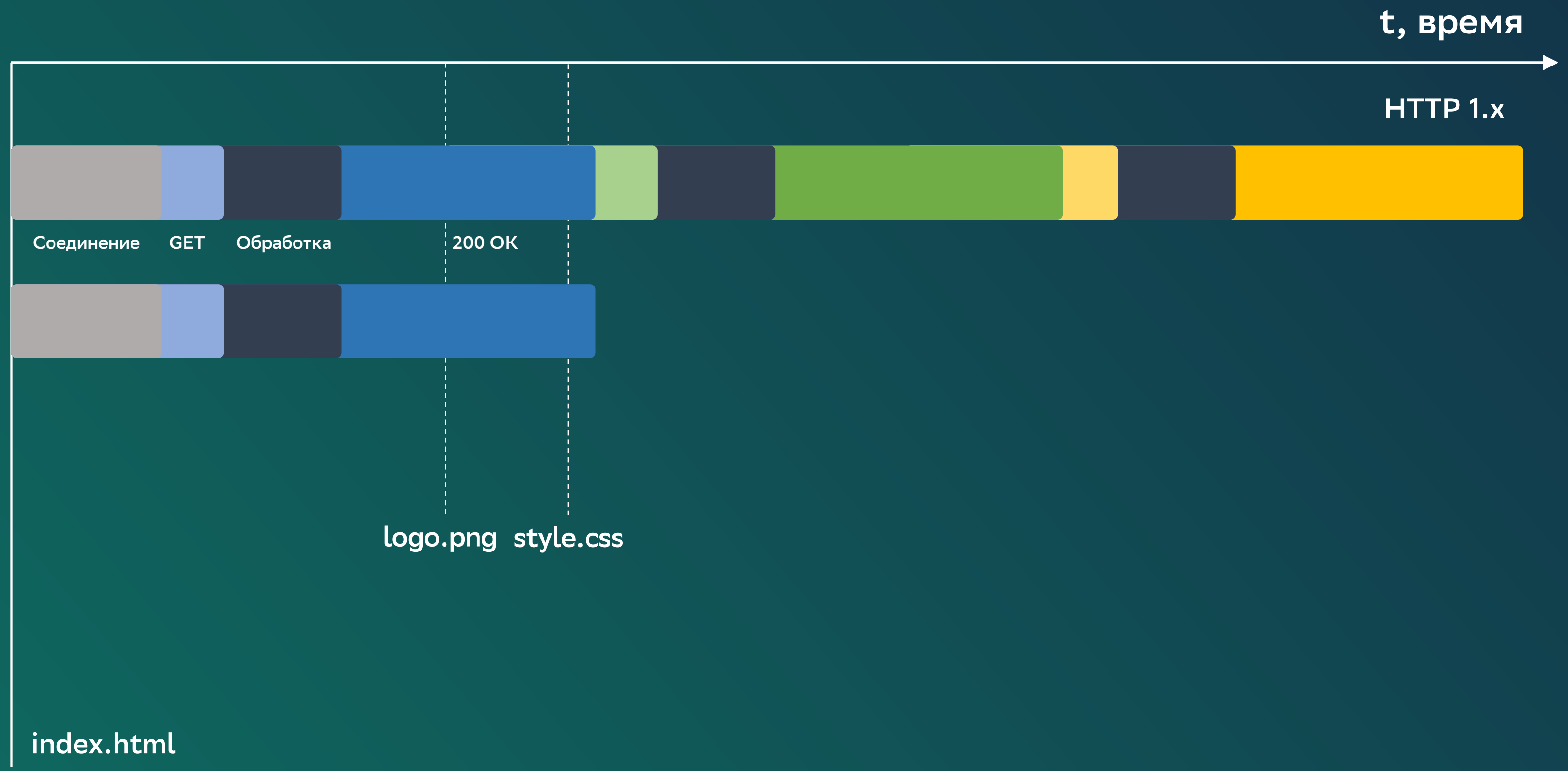
HTTP


t, время
HTTP 1.x
Соединение
GET
Обработка
200 OK
logo.png
style.css
index.html

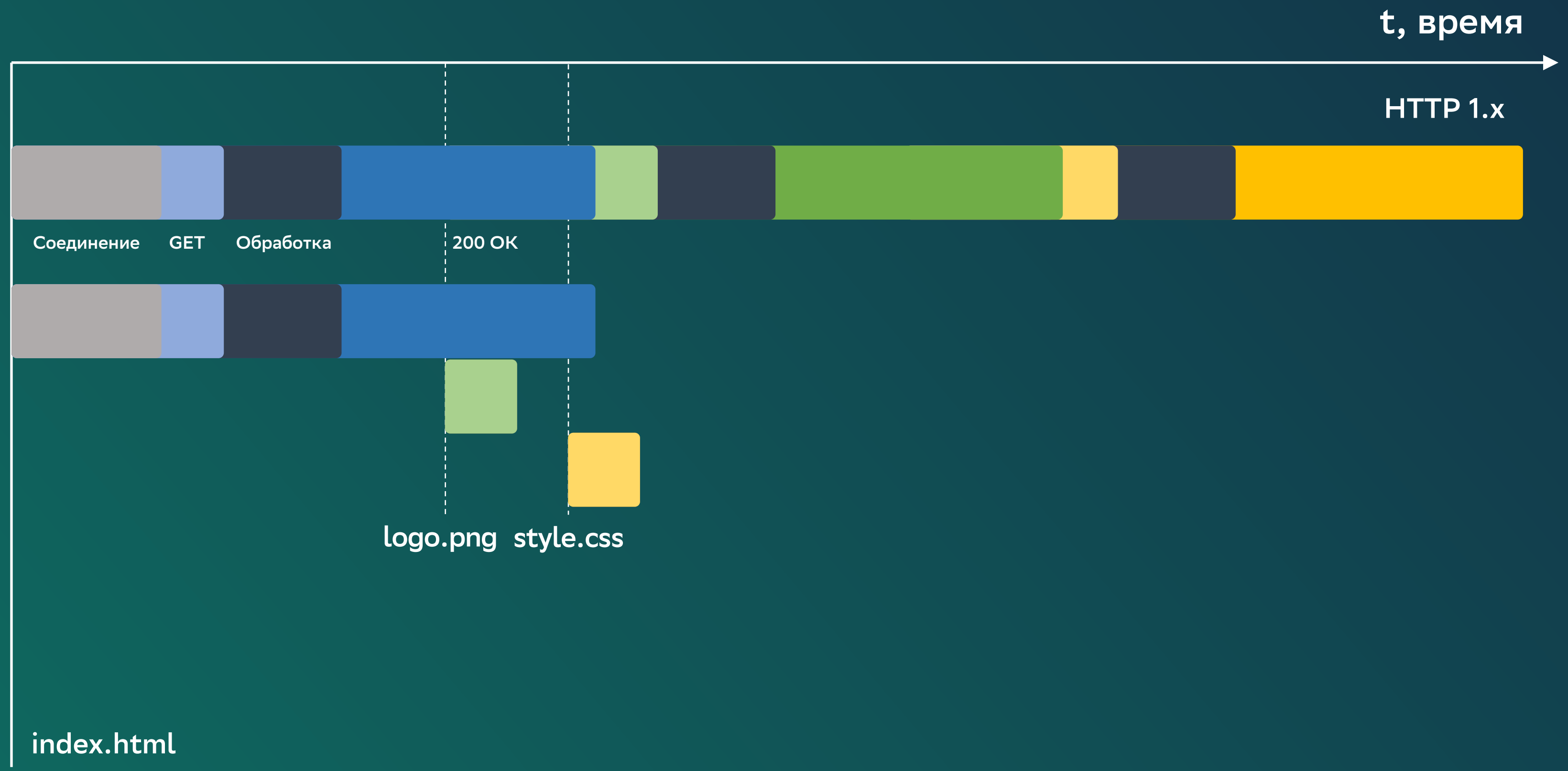
HTTP


t, время
HTTP 1.x
Соединение
GET
Обработка
200 OK
logo.png
style.css
index.html

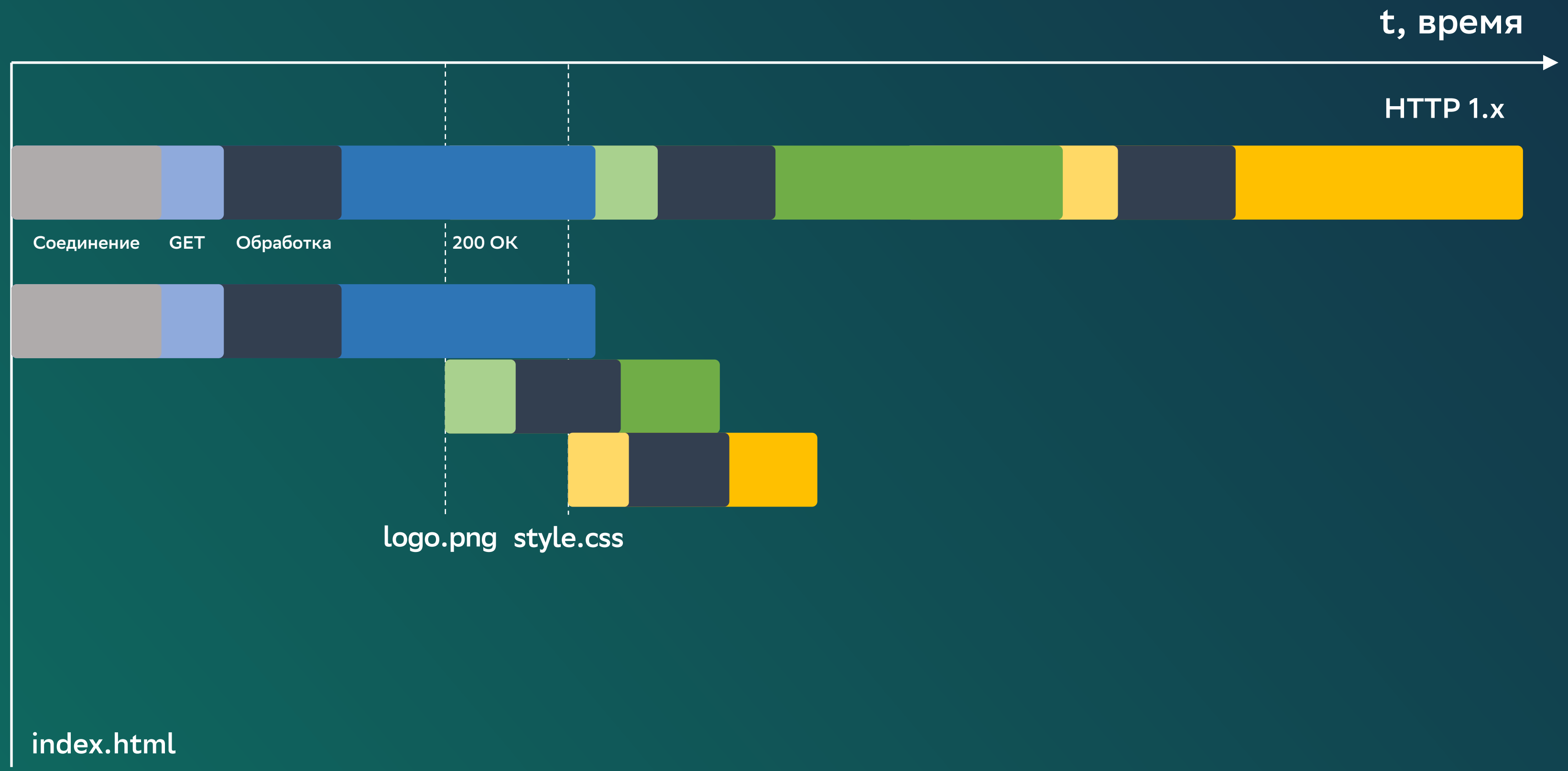
HTTP


t, время
HTTP 1.x
Соединение
GET
Обработка
200 OK
logo.png
style.css
index.html

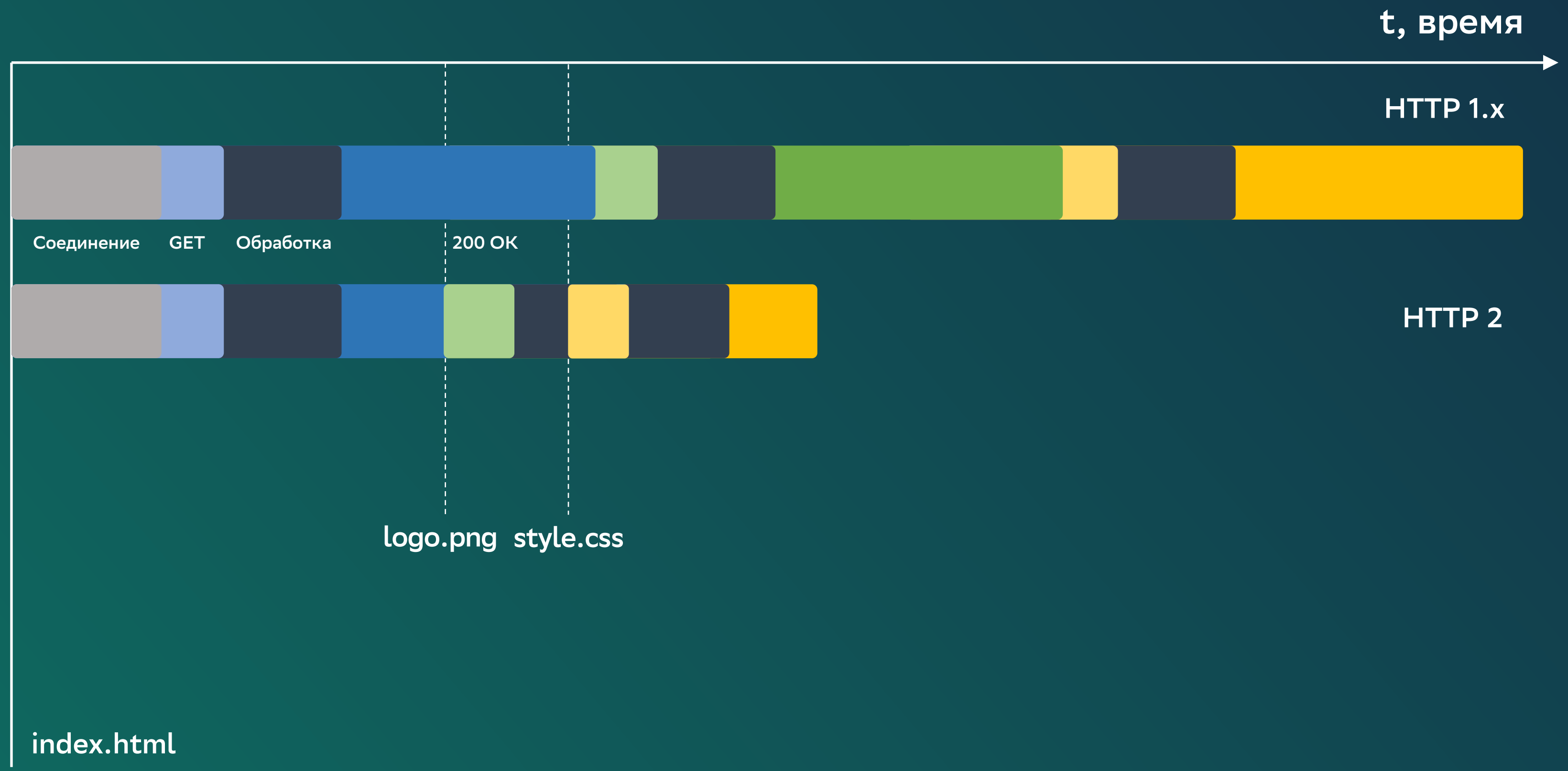
HTTP


t, время
HTTP 1.x
Соединение
GET
Обработка
200 OK
logo.png
style.css
index.html

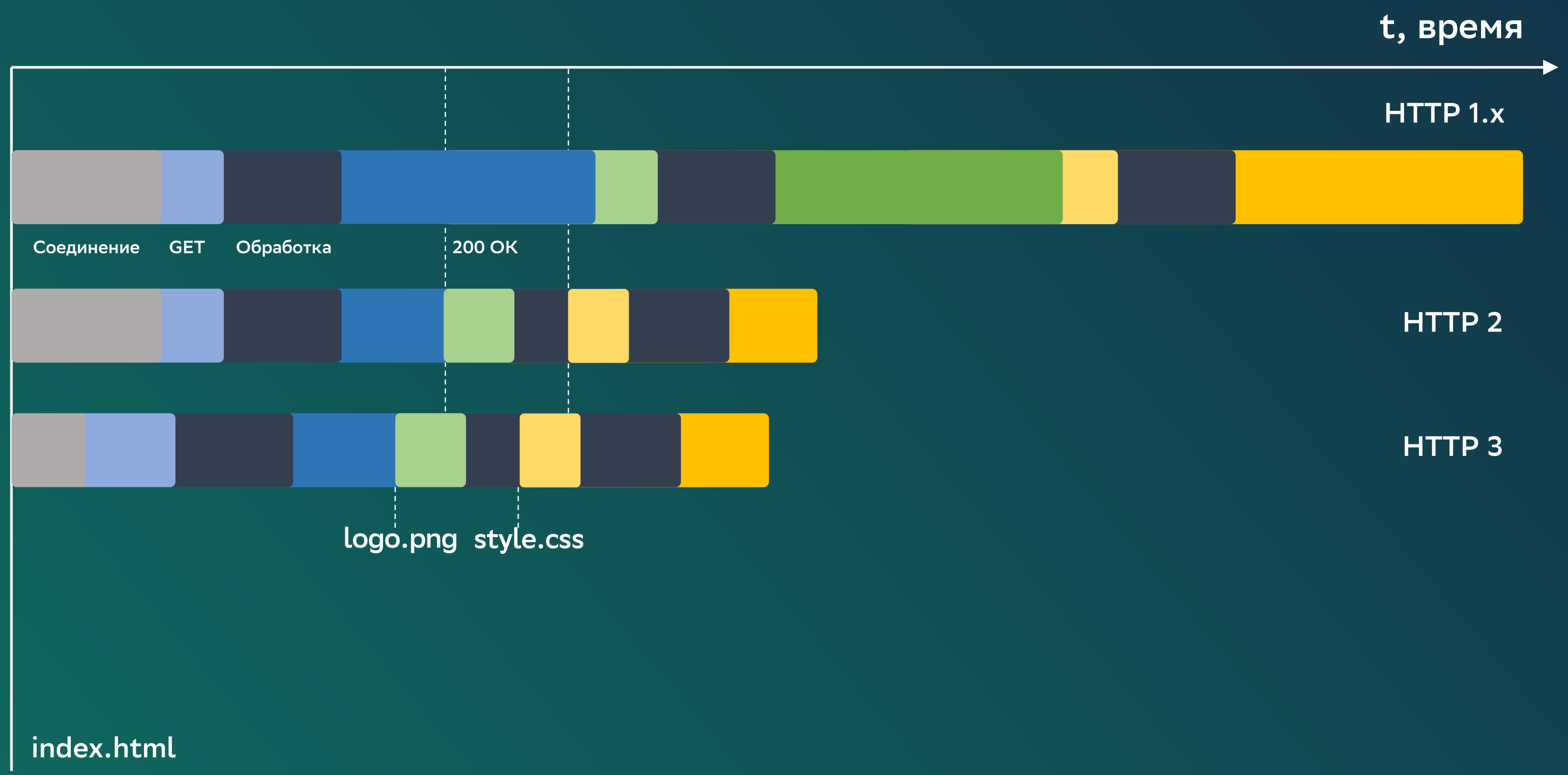
HTTP


t, время
HTTP 1.x
Соединение
GET
Обработка
200 OK
logo.png
style.css
index.html

HTTP

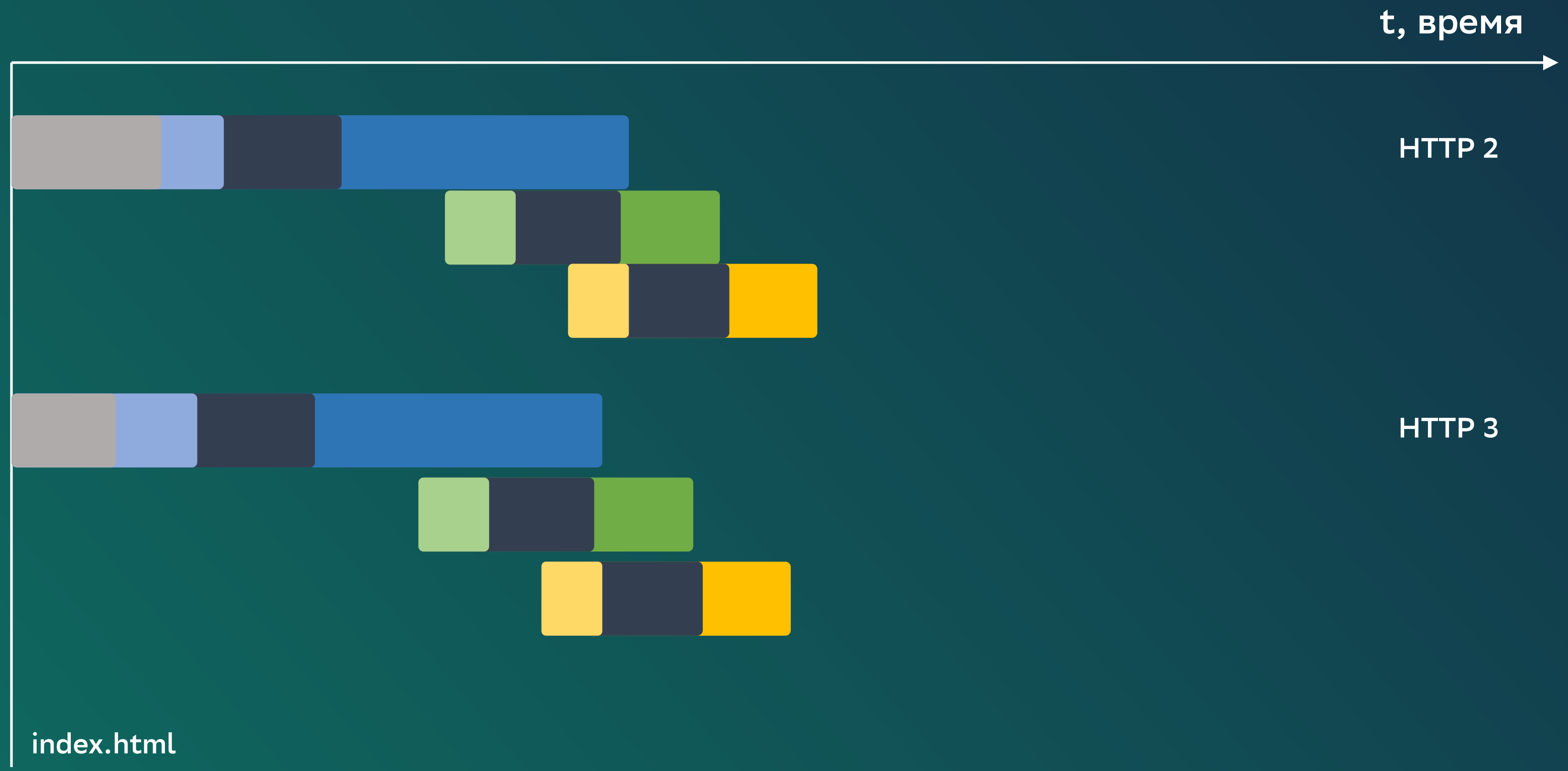

t, время
HTTP 1.x
Соединение
GET
Обработка
200 OK
HTTP 2
logo.png
style.css
index.html

HTTP

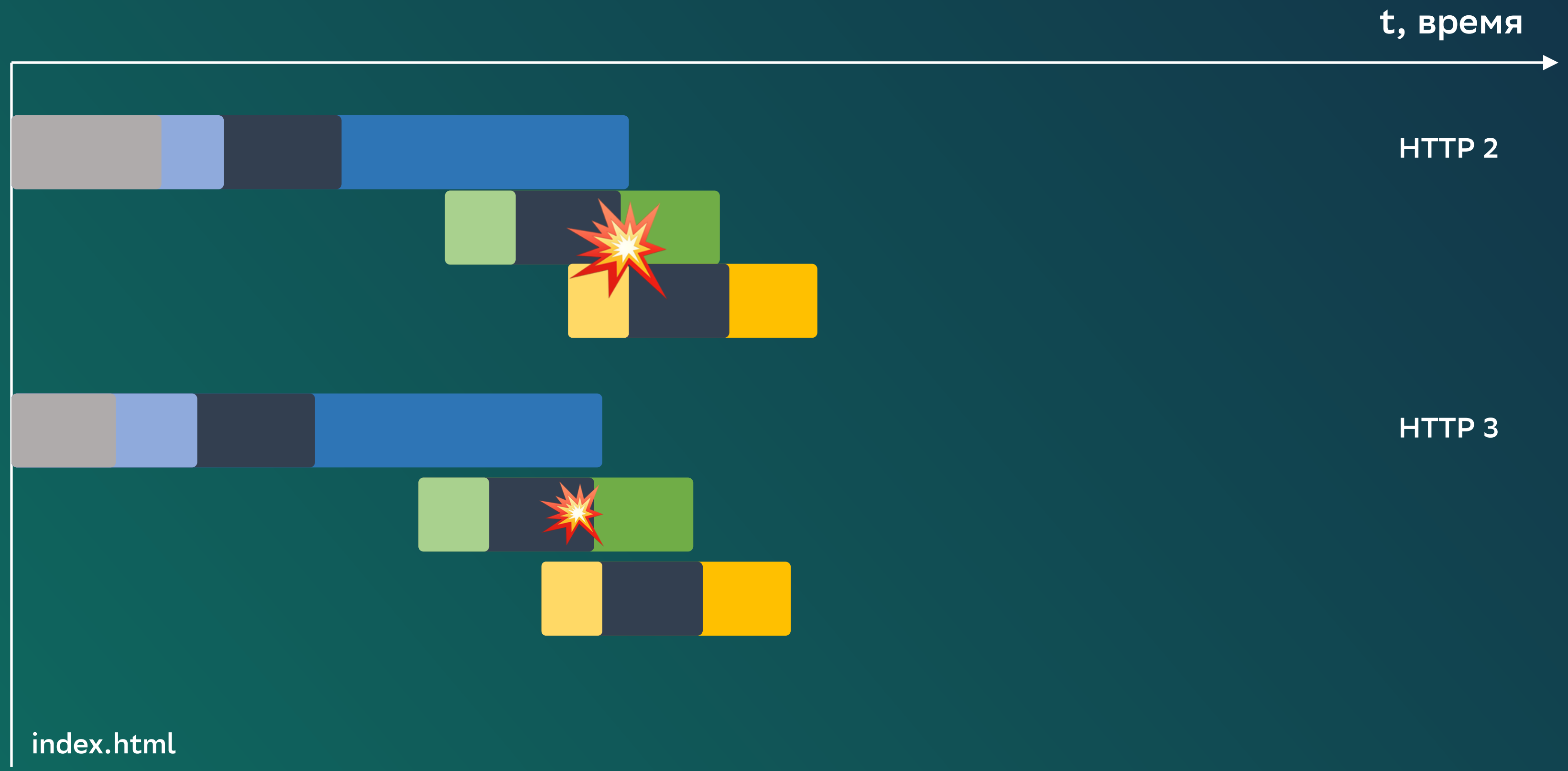


t, время
HTTP 1.x
Соединение
GET
Обработка
200 OK
HTTP 2
HTTP 3
logo.png
style.css
index.html

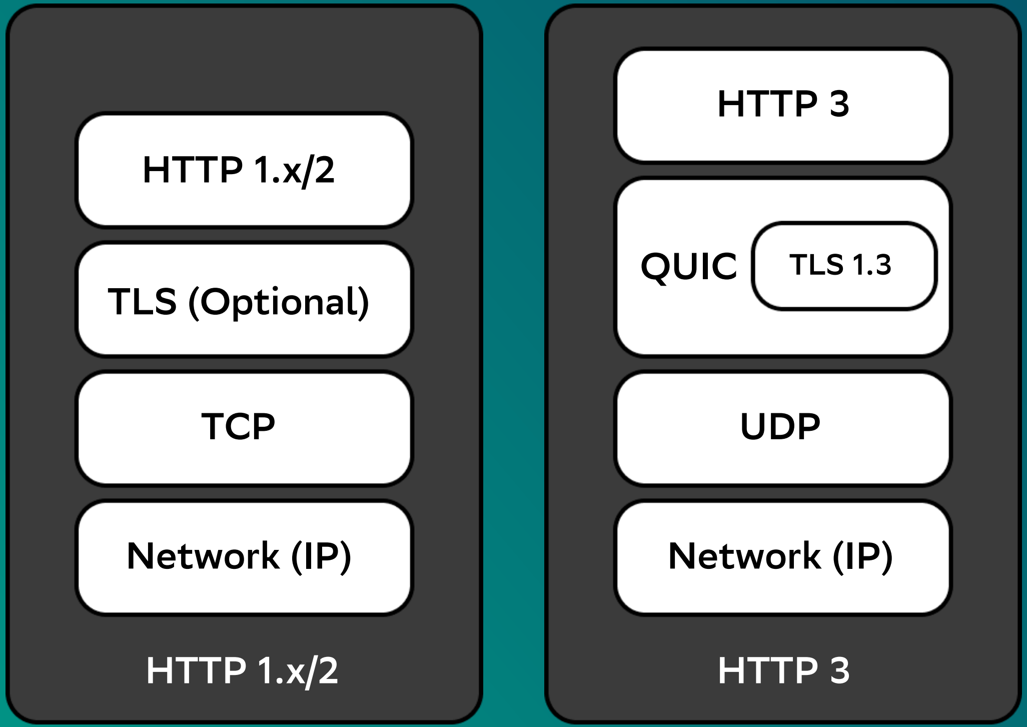
HTTP2 vs HTTP3



HTTP2 vs HTTP3



HTTP3. Мотивация



Протокол TCP не знает о независимости потоков в HTTP/2, и это означает:

проблема HOL blocking на уровне TCP из-за потерянных или задержанных пакетов превращается в проблему блокировки начала очереди для HTTP.

QUIC. Преимущества

Установление соединения происходит быстрее

Улучшенная обработка ошибок

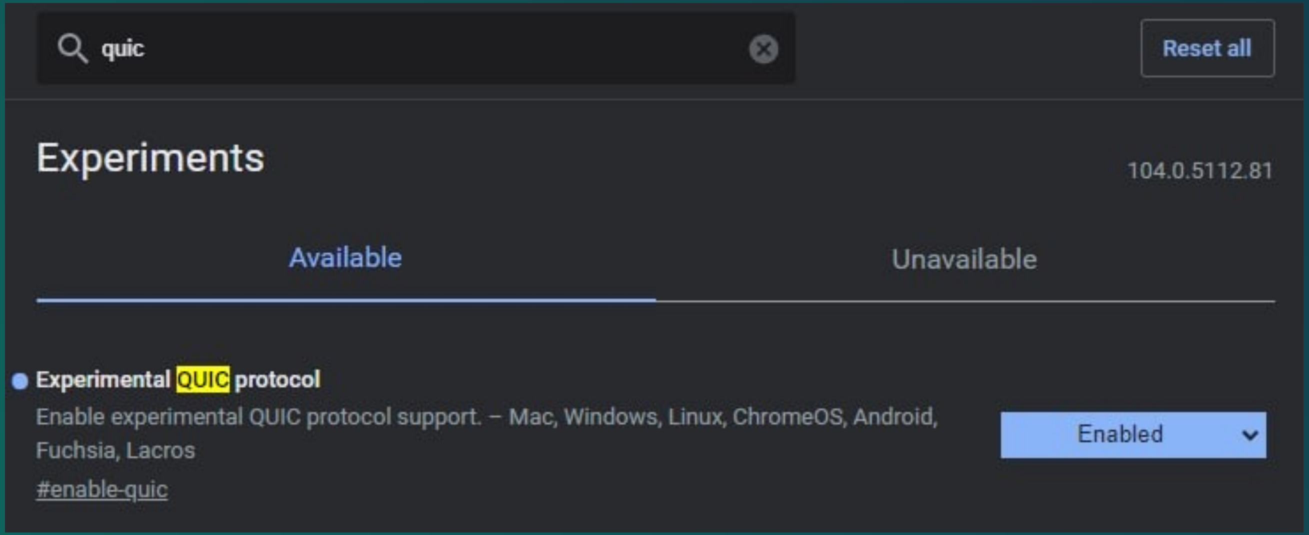
Переключение между сетями (Connection migration)

Как включить?

HTTP2 vs HTTP3

Как включить?

chrome://flags/



Experiments

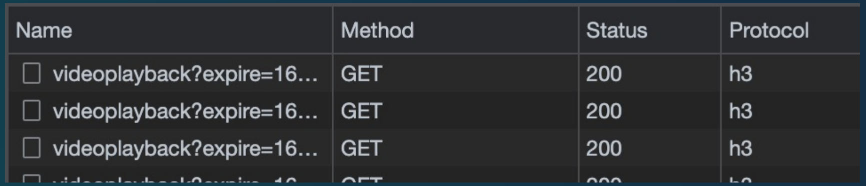
104.0.5112.81

● **Experimental QUIC protocol**

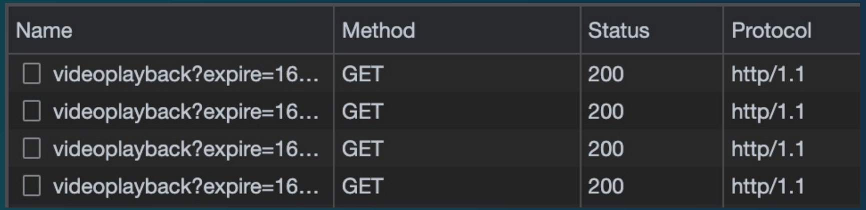
Enable experimental QUIC protocol support. – Mac, Windows, Linux, ChromeOS, Android, Fuchsia, Lacros

#enable-quic

Enabled



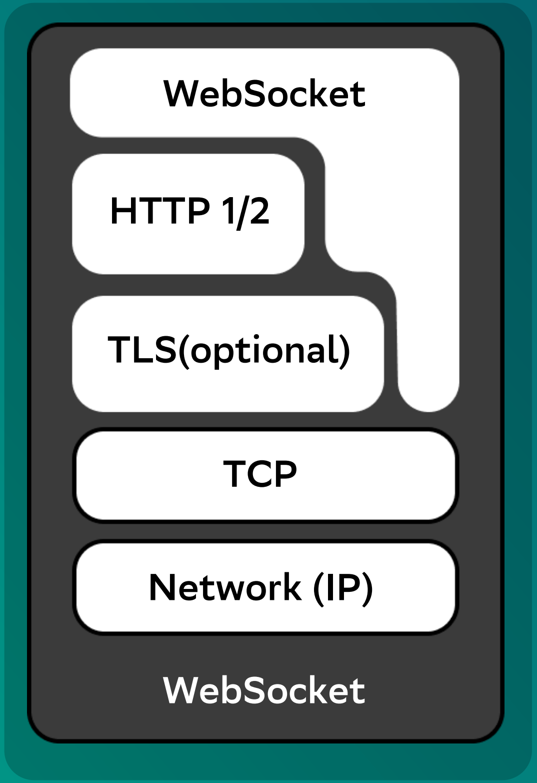
Name	Method	Status	Protocol
☐ videoplayback?expire=16…	GET	200	h3
☐ videoplayback?expire=16…	GET	200	h3
☐ videoplayback?expire=16…	GET	200	h3



Name	Method	Status	Protocol
☐ videoplayback?expire=16…	GET	200	http/1.1
☐ videoplayback?expire=16…	GET	200	http/1.1
☐ videoplayback?expire=16…	GET	200	http/1.1
☐ videoplayback?expire=16…	GET	200	http/1.1

Протоколы

WebSocket



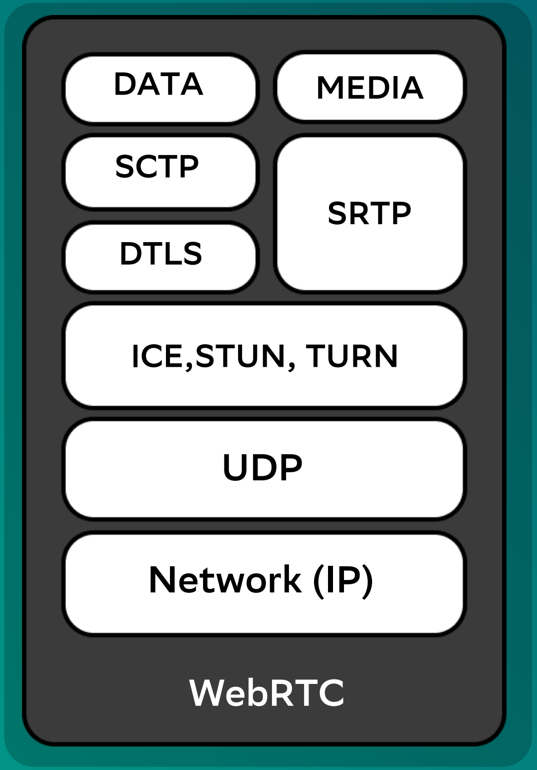
WebSocket
TCP
Head-of-line blocking

Head-of-line blocking из-за использования TCP

Отсутствует автоматическое восстановление при разрыве соединения

Потенциальный переезд на http/3 аннулируется наличием первичного handshake на соединении

WebRTC



DATA

MEDIA

SCTP

SRTP

DTLS

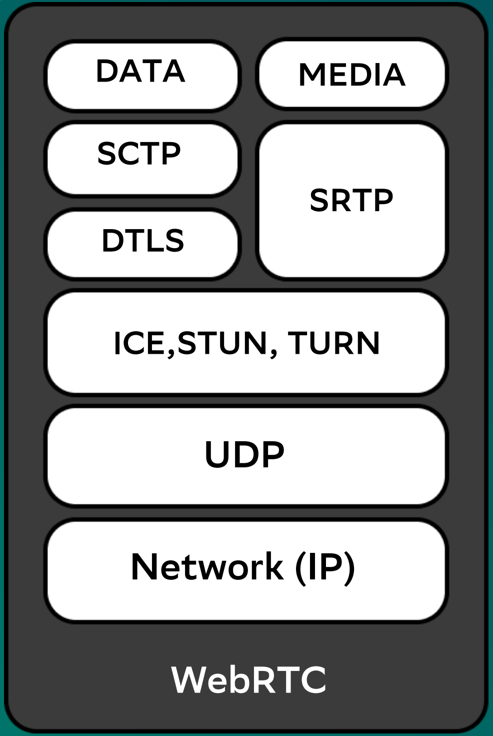
ICE,STUN, TURN

UDP

Network (IP)

WebRTC

WebRTC

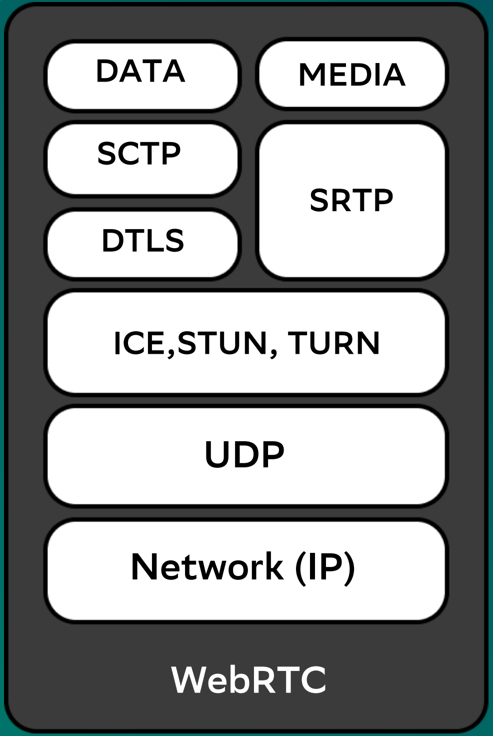


Тесно связанный стек различных протоколов

Не всегда необходимые p2p соединения

«Экзотичные» протоколы для транспорта и безопасности (ICE, DTLS, SCTP)

WebRTC



DATA

MEDIA

SCTP

SRTP

DTLS

ICE,STUN, TURN

UDP

Network (IP)

WebRTC

Для специализированных Real Time приложений нужны:

Поддержка кодеков

WebCodecs

Вычислительная логика

WebAssembly

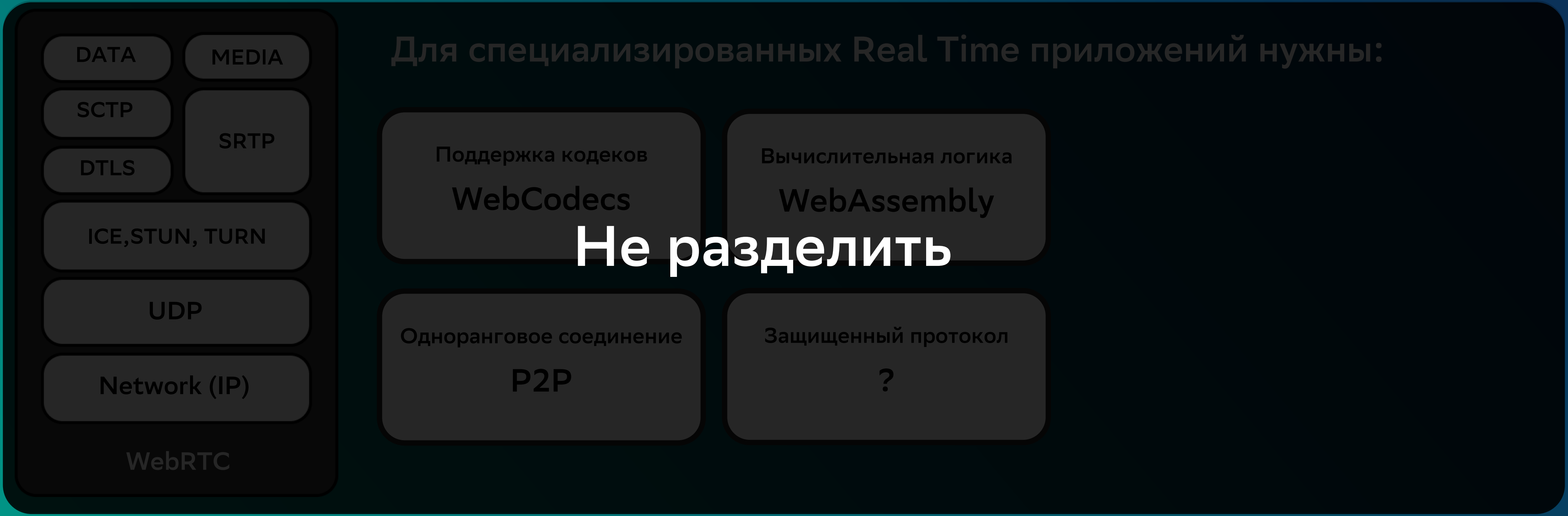
Одноранговое соединение

P2P

Защищенный протокол

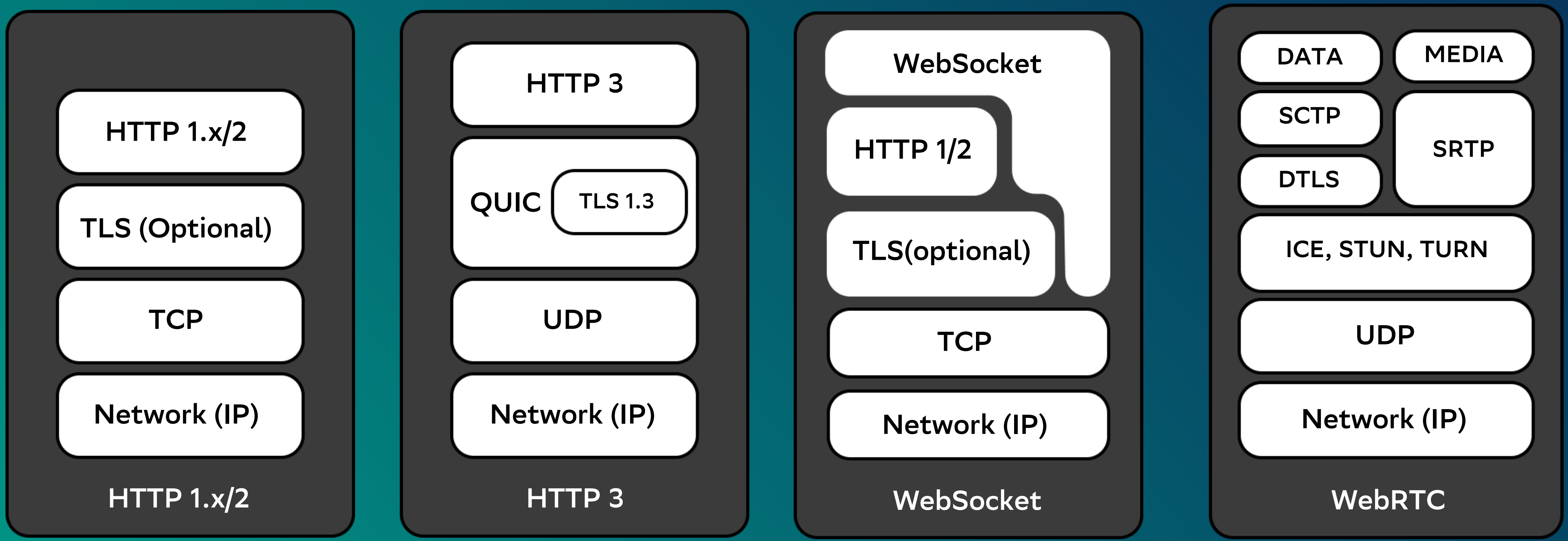
?

WebRTC



Не разделить

Протоколы


HTTP 1.x/2
TLS (Optional)
TCP
Network (IP)
HTTP 1.x/2
HTTP 3
QUIC
TLS 1.3
UDP
Network (IP)
HTTP 3
WebSocket
HTTP 1/2
TLS(optional)
TCP
Network (IP)
WebSocket
DATA
MEDIA
SCTP
DTLS
SRTP
ICE, STUN, TURN
UDP
Network (IP)
WebRTC

WebTransport

WebTransport

Преимущества:

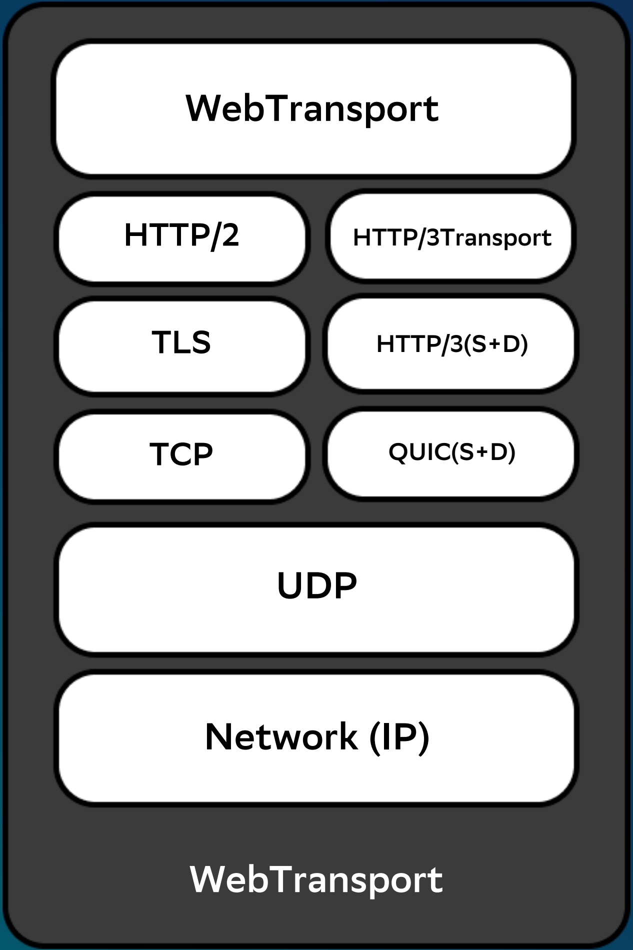
Работа в рамках модели веб-безопасности

Возможен фолбэк на HTTP/2

Работает в разных режимах

Использует Streams и Promises. Работает с async и await.

Может запускаться через WebWorkers



Датаграммы

TCP -> Пакеты

Упорядоченные, упакованные данные для сборки сообщений с проверкой целостности передачи

UDP -> Датаграммы

Поток данных без проверки на наличие ошибок и подтверждения доставки

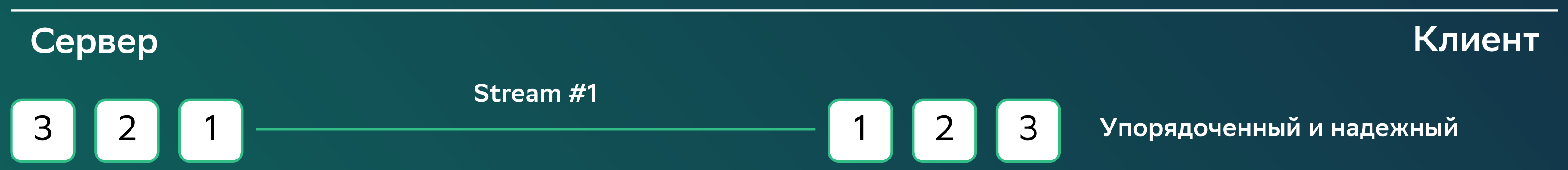
Режимы WebTransport

Режимы WebTransport

Сервер

Клиент

Режимы WebTransport



Сервер

Клиент

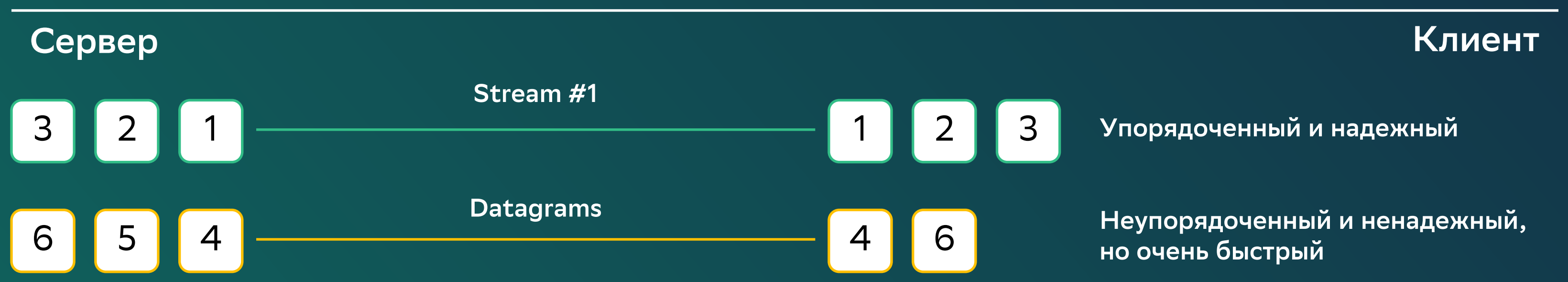
Stream #1

3 2 1

1 2 3

Упорядоченный и надежный

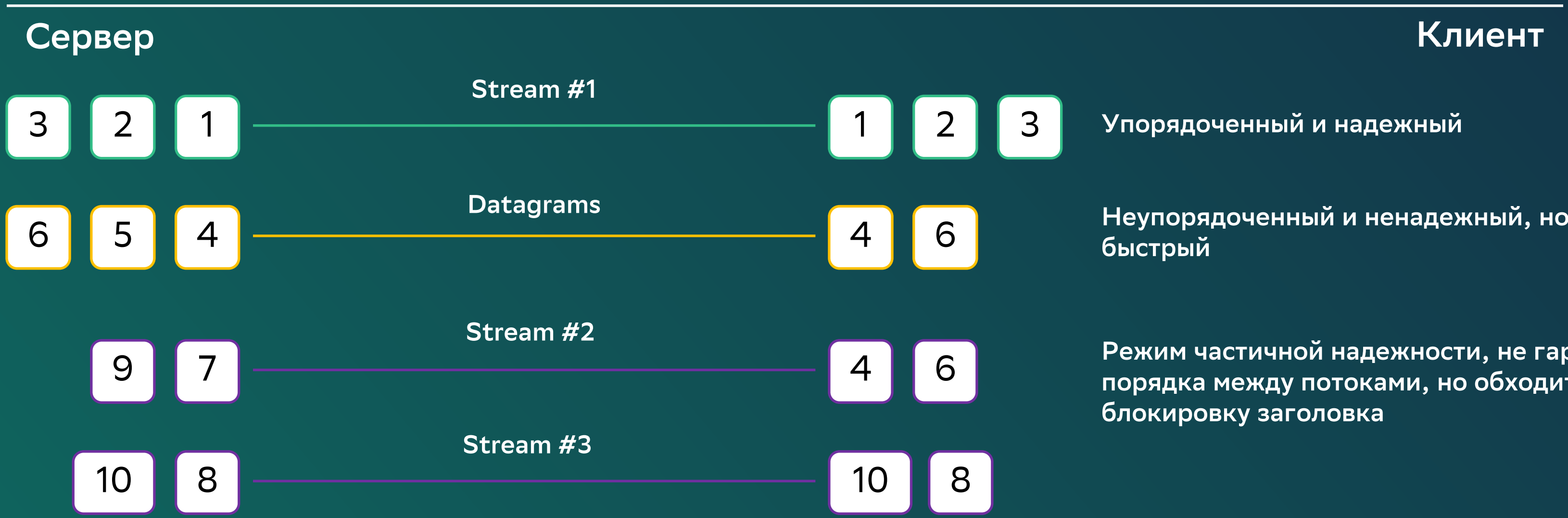
Режимы WebTransport



Упорядоченный и надежный

Неупорядоченный и ненадежный,
но очень быстрый

Режимы WebTransport

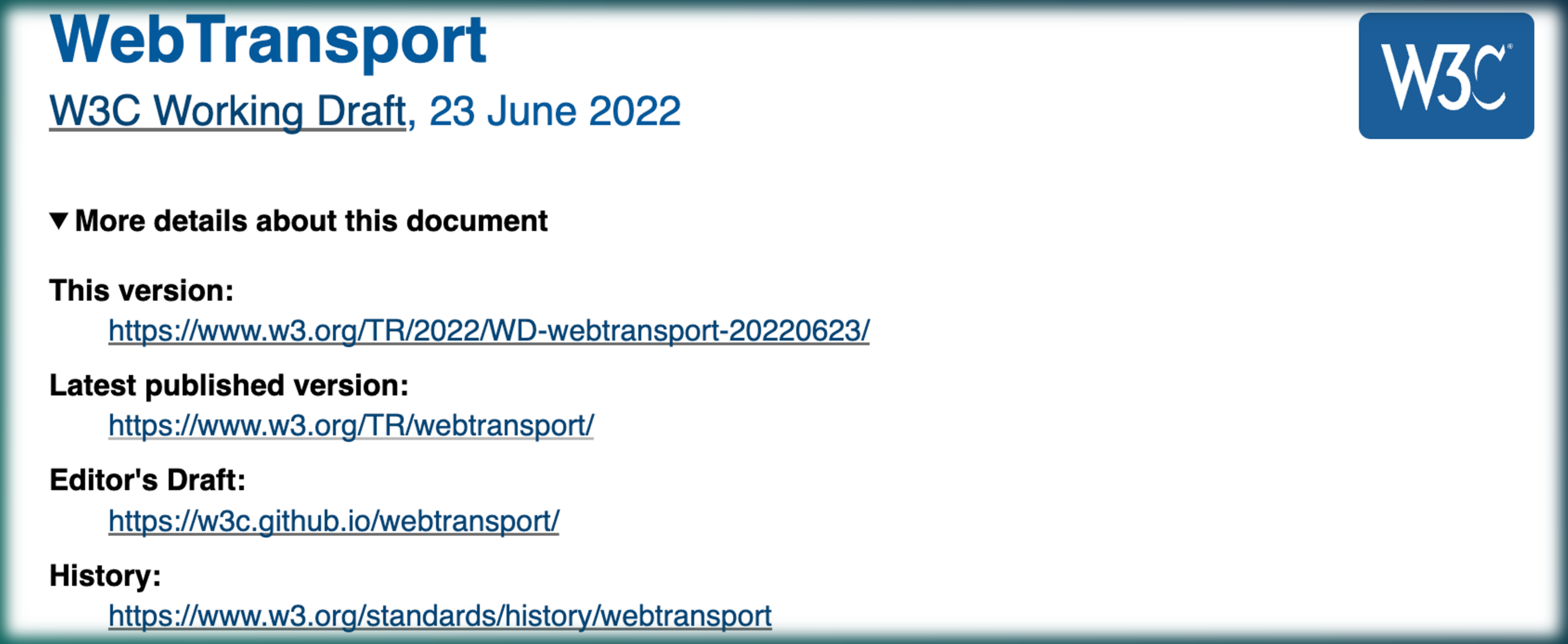


Упорядоченный и надежный

Неупорядоченный и ненадежный, но очень быстрый

Режим частичной надежности, не гарантирует порядка между потоками, но обходит блокировку заголовка

Статус WebTransport



WebTransport

W3C Working Draft, 23 June 2022

▼ More details about this document

This version:
https://www.w3.org/TR/2022/WD-webtransport-20220623/

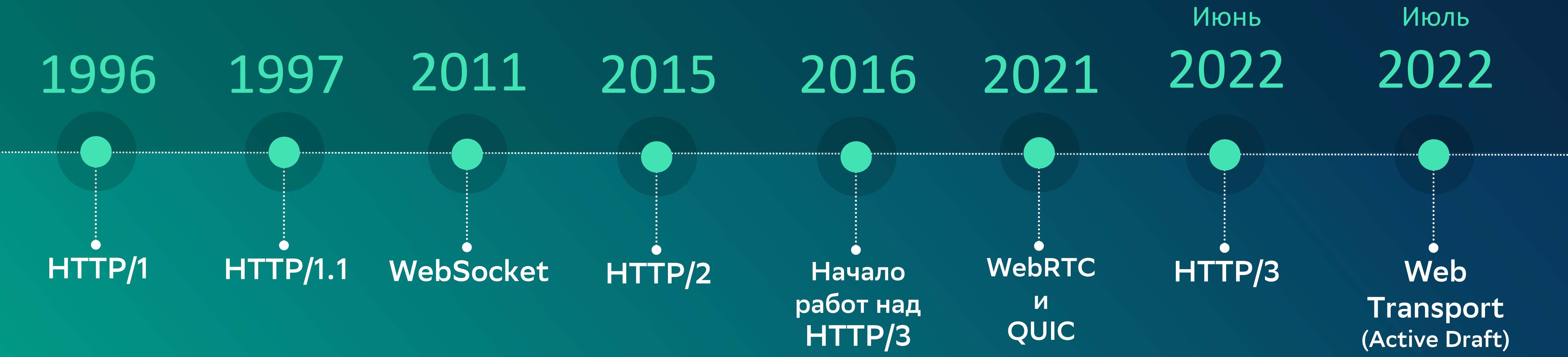
Latest published version:
https://www.w3.org/TR/webtransport/

Editor's Draft:
https://w3c.github.io/webtransport/

History:
https://www.w3.org/standards/history/webtransport

Текущий статус

Протоколы. История появления



1996 HTTP/1

1997 HTTP/1.1

2011 WebSocket

2015 HTTP/2

2016 Начало работ над HTTP/3

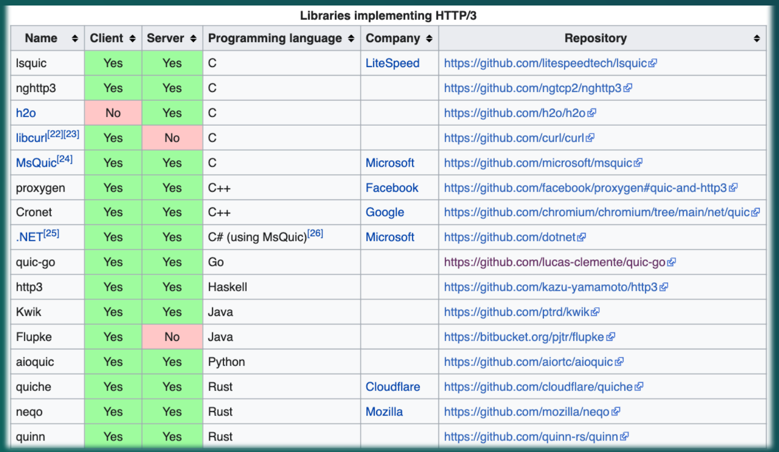
2021 WebRTC и QUIC

Июнь 2022 HTTP/3

Июль 2022 Web Transport (Active Draft)

QUIC, WebTransport. Текущий статус

QUIC



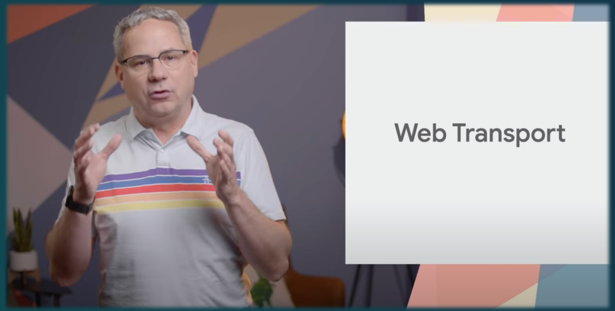
Libraries implementing HTTP/3

Name	Client	Server	Programming language	Company	Repository
lsquic	Yes	Yes	C	LiteSpeed	https://github.com/litespeedtech/lsquic
nghttp3	Yes	Yes	C		https://github.com/ngtcp2/nghttp3
h2o	No	Yes	C		https://github.com/h2o/h2o
libcurl[22][23]	Yes	No	C		https://github.com/curl/curl
MsQuic[24]	Yes	Yes	C	Microsoft	https://github.com/microsoft/msquic
proxygen	Yes	Yes	C++	Facebook	https://github.com/facebook/proxygen#quic-and-http3
Cronet	Yes	Yes	C++	Google	https://github.com/chromium/chromium/tree/main/net/quic
.NET[25]	Yes	Yes	C# (using MsQuic)[26]	Microsoft	https://github.com/dotnet
quic-go	Yes	Yes	Go		https://github.com/lucas-clemente/quic-go
http3	Yes	Yes	Haskell		https://github.com/kazu-yamamoto/http3
Kwik	Yes	Yes	Java		https://github.com/ptrd/kwik
Flupke	Yes	No	Java		https://bitbucket.org/pjtr/flupke
aioquic	Yes	Yes	Python		https://github.com/aiortc/aioquic
quiche	Yes	Yes	Rust	Cloudflare	https://github.com/cloudflare/quiche
neqo	Yes	Yes	Rust	Mozilla	https://github.com/mozilla/neqo
quinn	Yes	Yes	Rust		https://github.com/quinn-rs/quinn

Nginx. https://quic.nginx.org/

WebTransport

Сервера на Go и C#. Demo Client Python.



New in Chrome 97

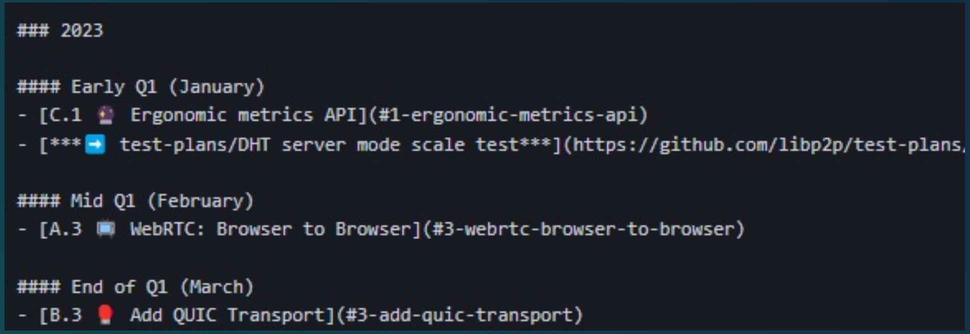
Node JS quic support

James M Snell

https://github.com/jasnell

Основная tracking задача по quic: https://github.com/nodejs/node/issues/38478

Roadmap



```
### 2023

#### Early Q1 (January)
- [C.1 🔮 Ergonomic metrics API](#1-ergonomic-metrics-api)
- [***➡️ test-plans/DHT server mode scale test***](https://github.com/libp2p/test-plans,

#### Mid Q1 (February)
- [A.3 📺 WebRTC: Browser to Browser](#3-webrtc-browser-to-browser)

#### End of Q1 (March)
- [B.3 🥊 Add QUIC Transport](#3-add-quic-transport)
```

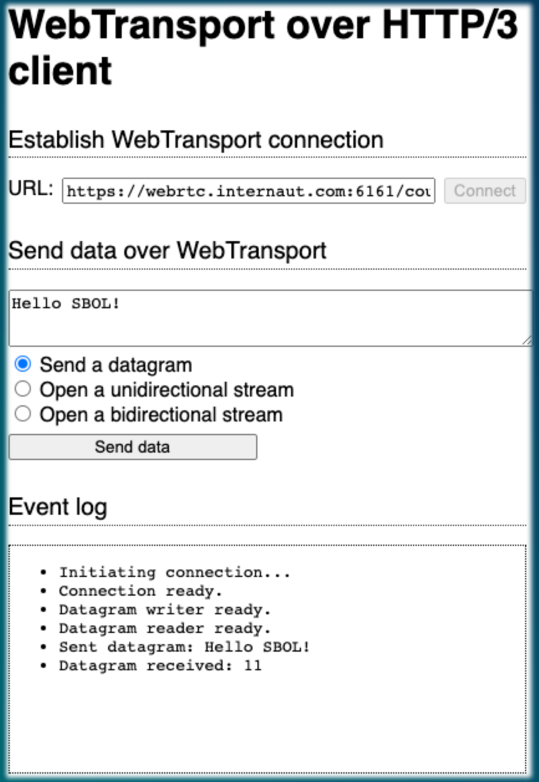
Где потрогать?

Клиент (локально):

WebTransport Python client Demo - https://github.com/w3c/webtransport/tree/main/samples/echo

Клиент(в web):

Chrome - https://webrtc.internaut.com/wt/



WebTransport over HTTP/3 client

Establish WebTransport connection

URL: https://webrtc.internaut.com:6161/cou

Send data over WebTransport

Hello SBOL!

● Send a datagram
○ Open a unidirectional stream
○ Open a bidirectional stream

Event log

```
• Initiating connection...
• Connection ready.
• Datagram writer ready.
• Datagram reader ready.
• Sent datagram: Hello SBOL!
• Datagram received: 11
```

Не всегда работает

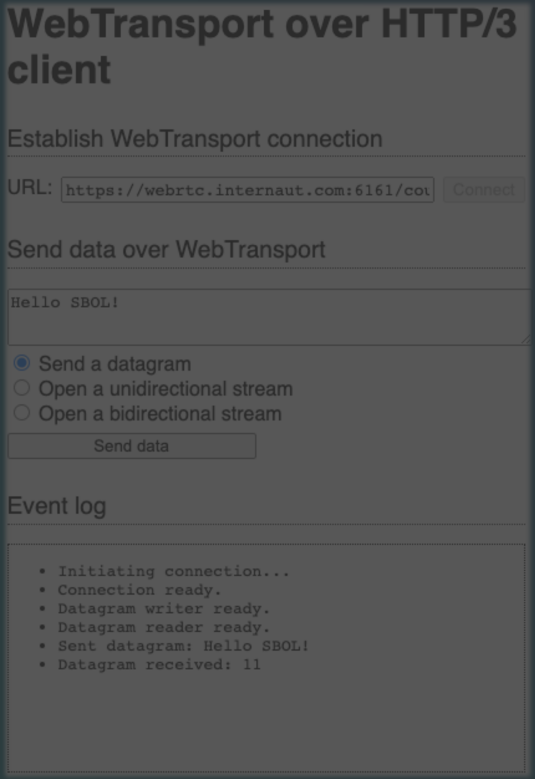
Где потрогать?

Клиент (локально):

WebTransport Python client Demo - https://github.com/w3c/webtransport/tree/main/samples/echo

Клиент(в web):

Chrome - https://webrtc.internaut.com/wt/



WebTransport over HTTP/3 client

Establish WebTransport connection

URL: https://webrtc.internaut.com:6161/cou

Send data over WebTransport

Hello SBOL!

● Send a datagram
○ Open a unidirectional stream
○ Open a bidirectional stream

Event log

```
• Initiating connection...
• Connection ready.
• Datagram writer ready.
• Datagram reader ready.
• Sent datagram: Hello SBOL!
• Datagram received: 11
```

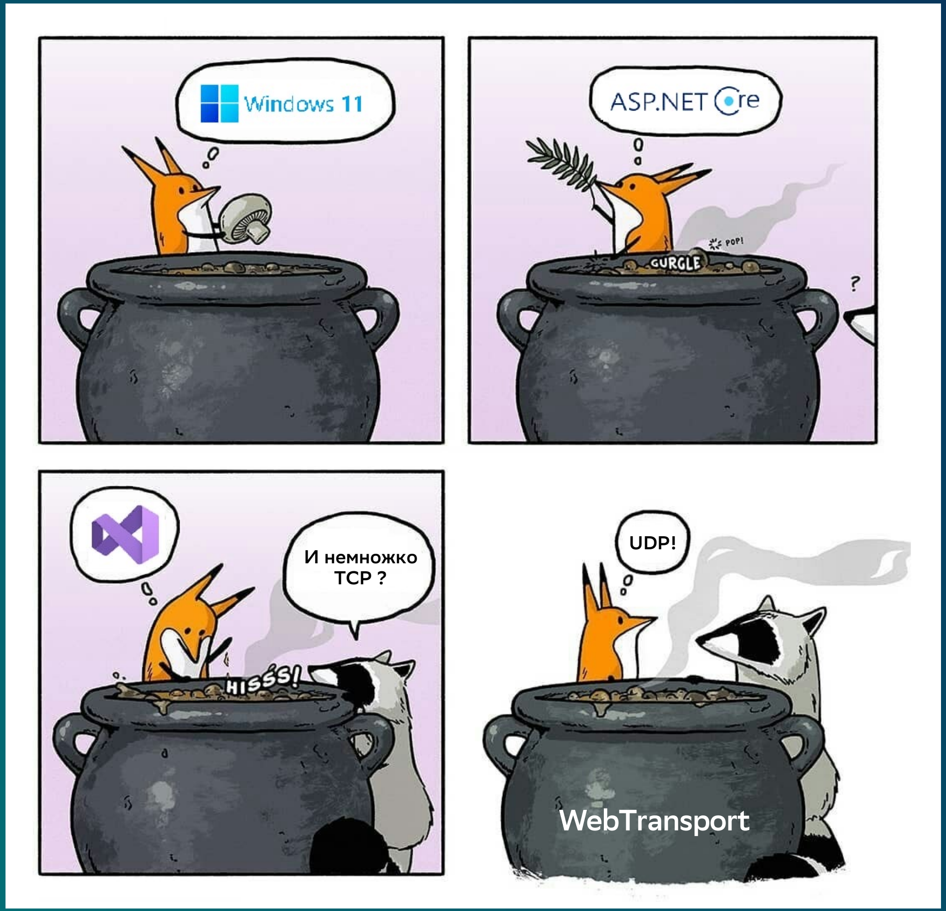
Готовим WebTransport

Технологии

Windows 11 сборка 22000+

ASP NET CORE 6+

Visual studio 2022



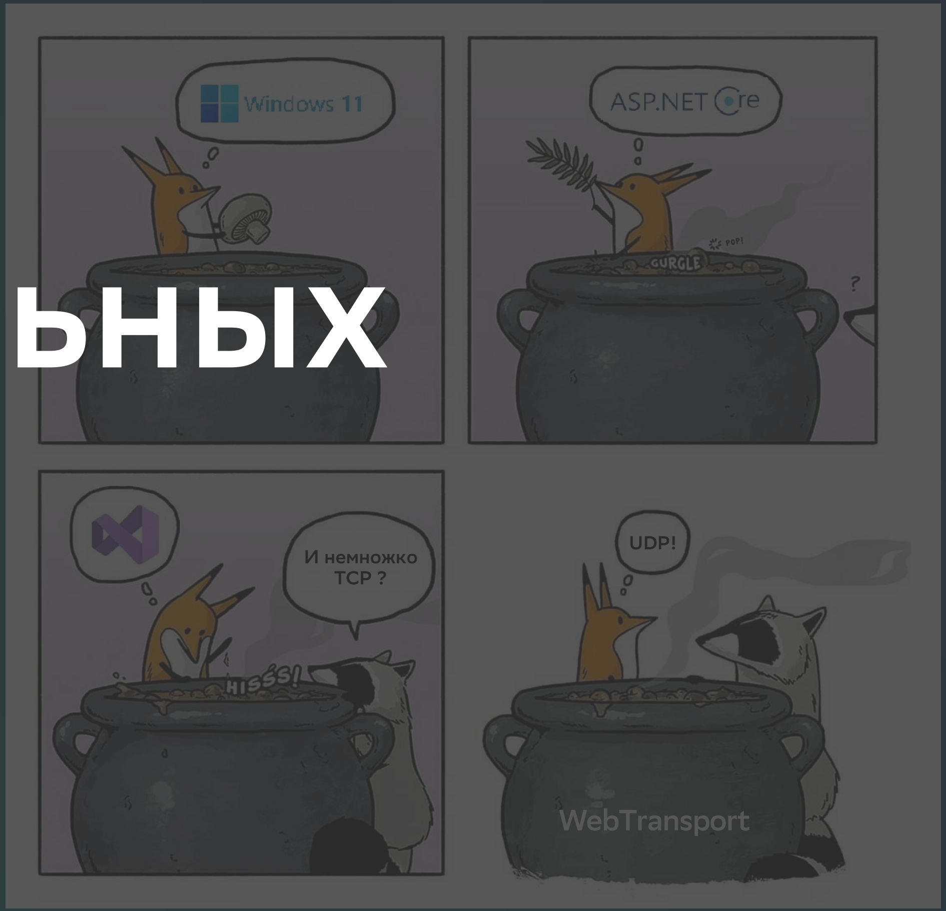
Нет официальных рецептов

Технологии

Windows 11 сборка 22000+

ASP.NET CORE 6+

Visual studio 2022

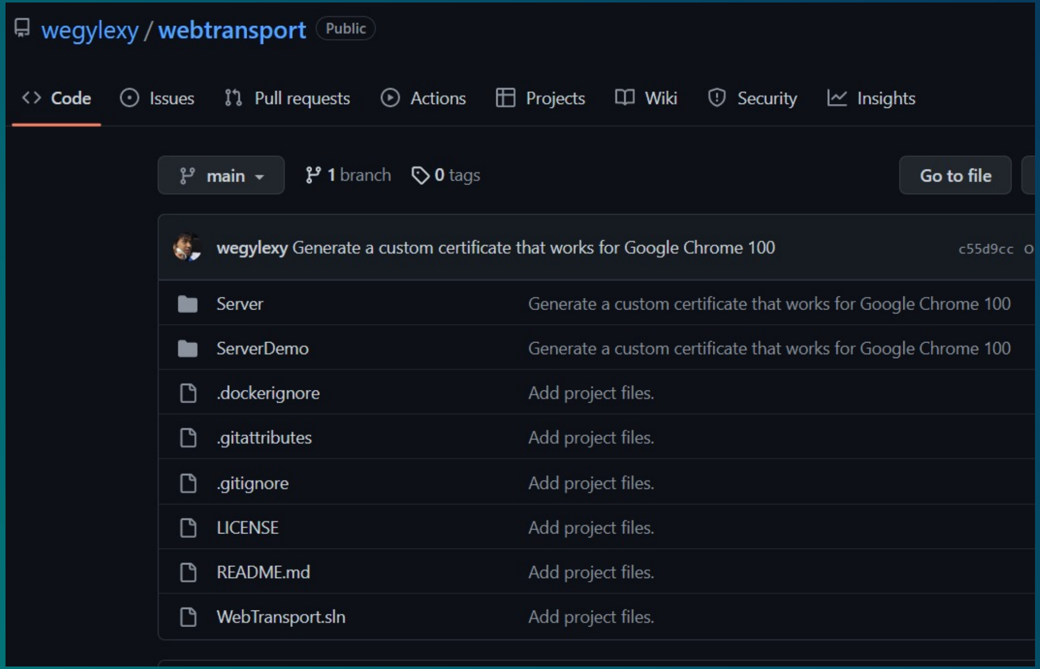


Ищи на github

Готовим WebTransport

https://github.com/wegylexy

https://github.com/wegylexy/webtransport



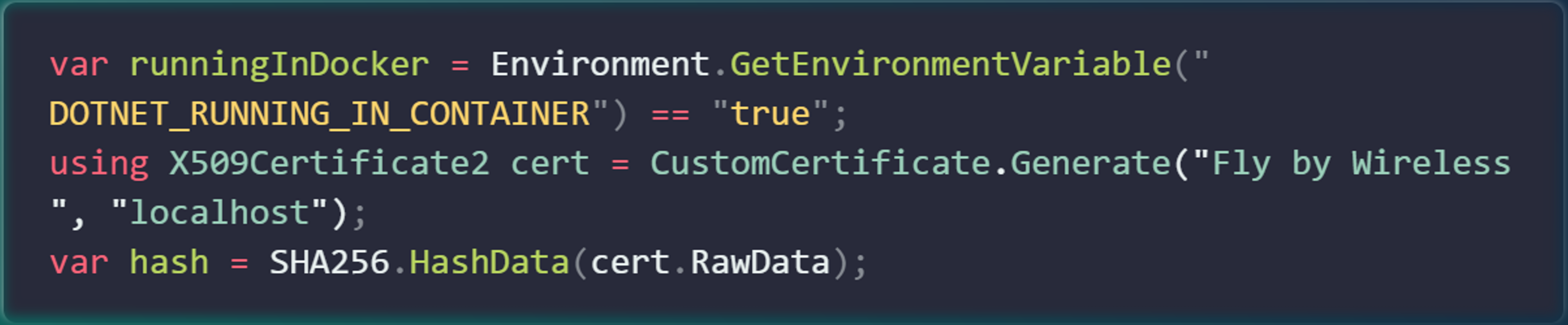
wegylexy Generate a custom certificate that works for Google Chrome 100 c55d9cc

Server	Generate a custom certificate that works for Google Chrome 100
ServerDemo	Generate a custom certificate that works for Google Chrome 100
.dockerignore	Add project files.
.gitattributes	Add project files.
.gitignore	Add project files.
LICENSE	Add project files.
README.md	Add project files.
WebTransport.sln	Add project files.

Готовим WebTransport

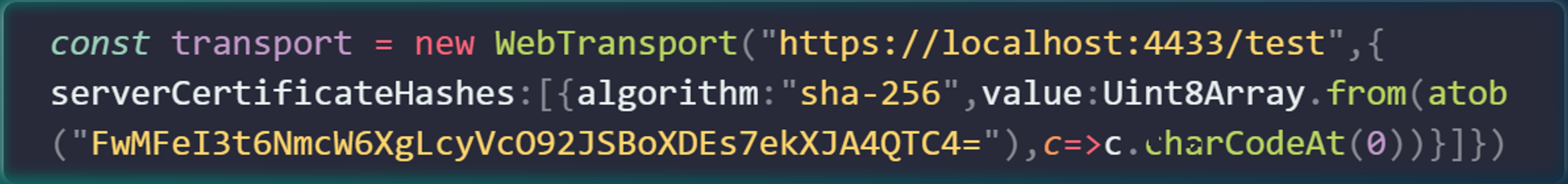
Открываем файл ServerDemo\Program.cs

Готовим WebTransport



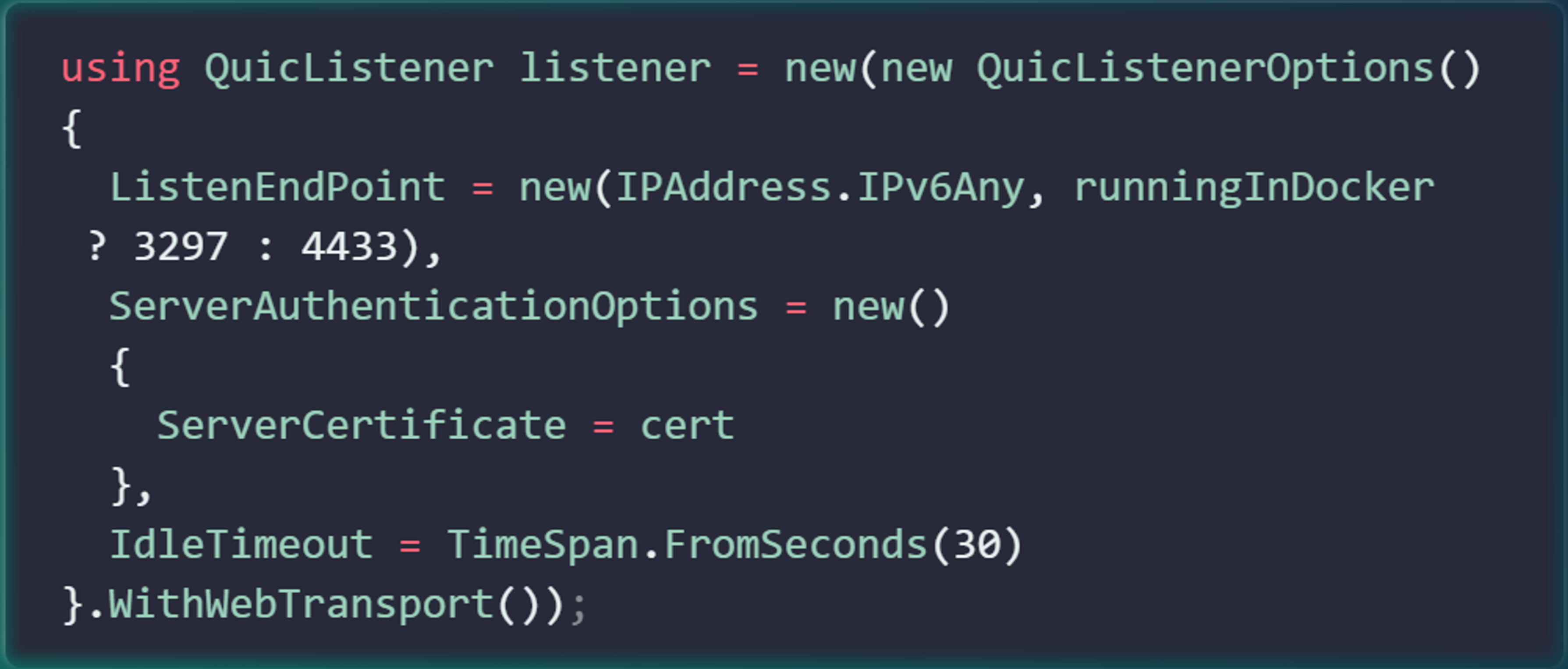
```
var runningInDocker = Environment.GetEnvironmentVariable("
DOTNET_RUNNING_IN_CONTAINER") == "true";
using X509Certificate2 cert = CustomCertificate.Generate("Fly by Wireless
", "localhost");
var hash = SHA256.HashData(cert.RawData);
```

Готовим WebTransport



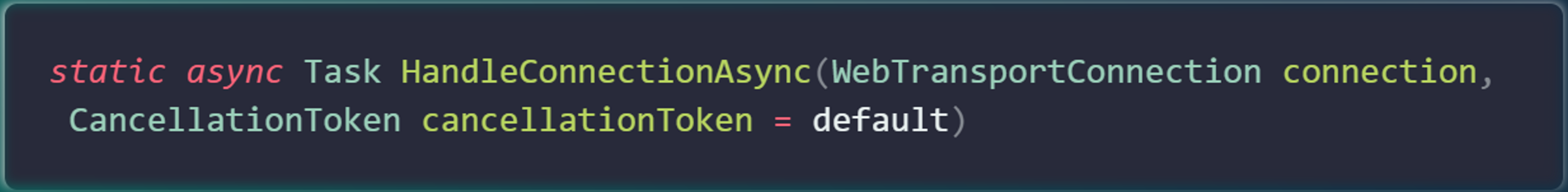
```
const transport = new WebTransport("https://localhost:4433/test",{
serverCertificateHashes:[{algorithm:"sha-256",value:Uint8Array.from(atob
("FwMFeI3t6NmcW6XgLcyVcO92JSBoXDEs7ekXJA4QTC4="),c=>c.charCodeAt(0))}]})
```

Готовим WebTransport



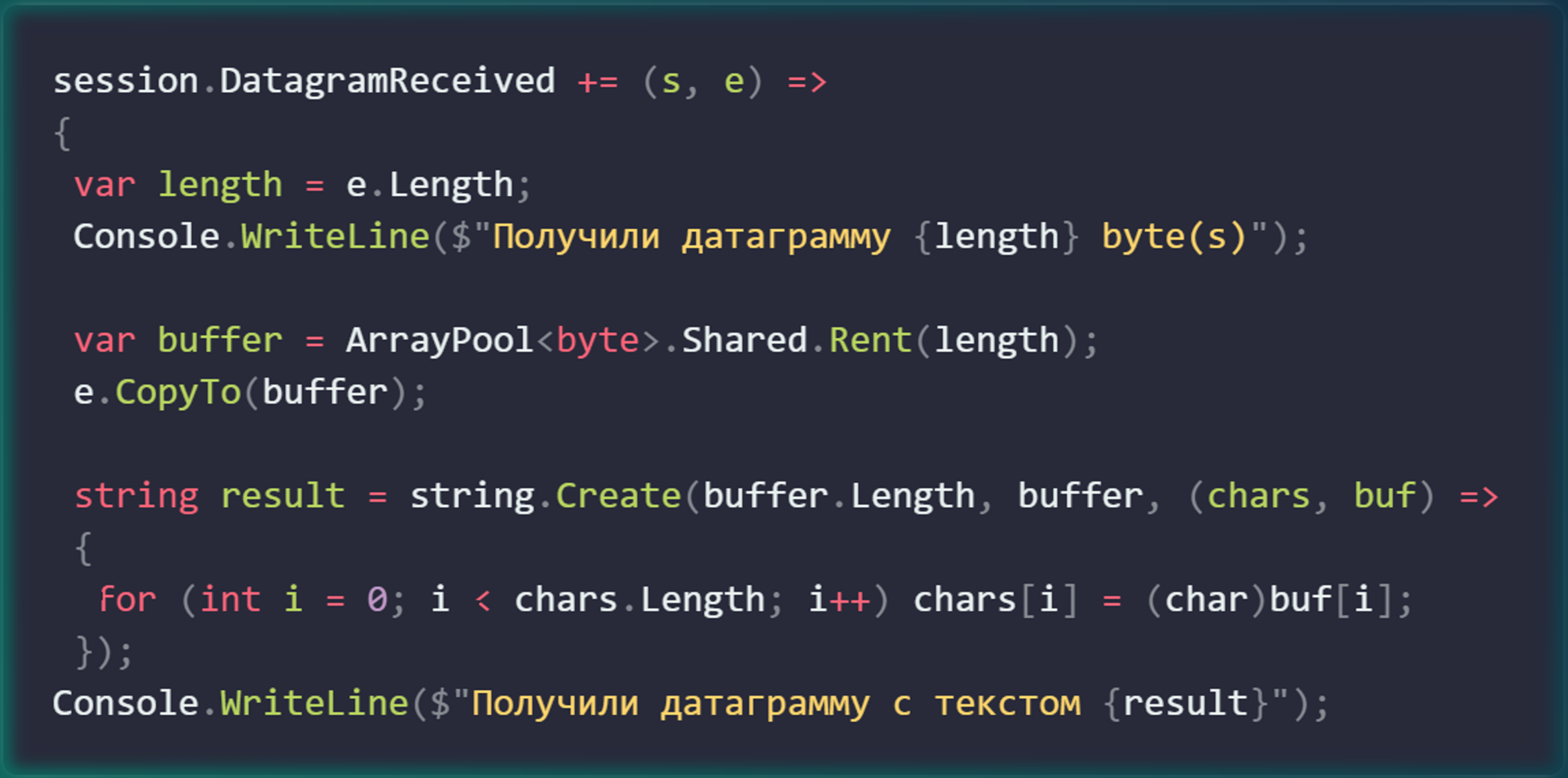
```
using QuicListener listener = new(new QuicListenerOptions()
{
  ListenEndPoint = new(IPAddress.IPv6Any, runningInDocker
 ? 3297 : 4433),
  ServerAuthenticationOptions = new()
  {
    ServerCertificate = cert
  },
  IdleTimeout = TimeSpan.FromSeconds(30)
}.WithWebTransport());
```

Готовим WebTransport



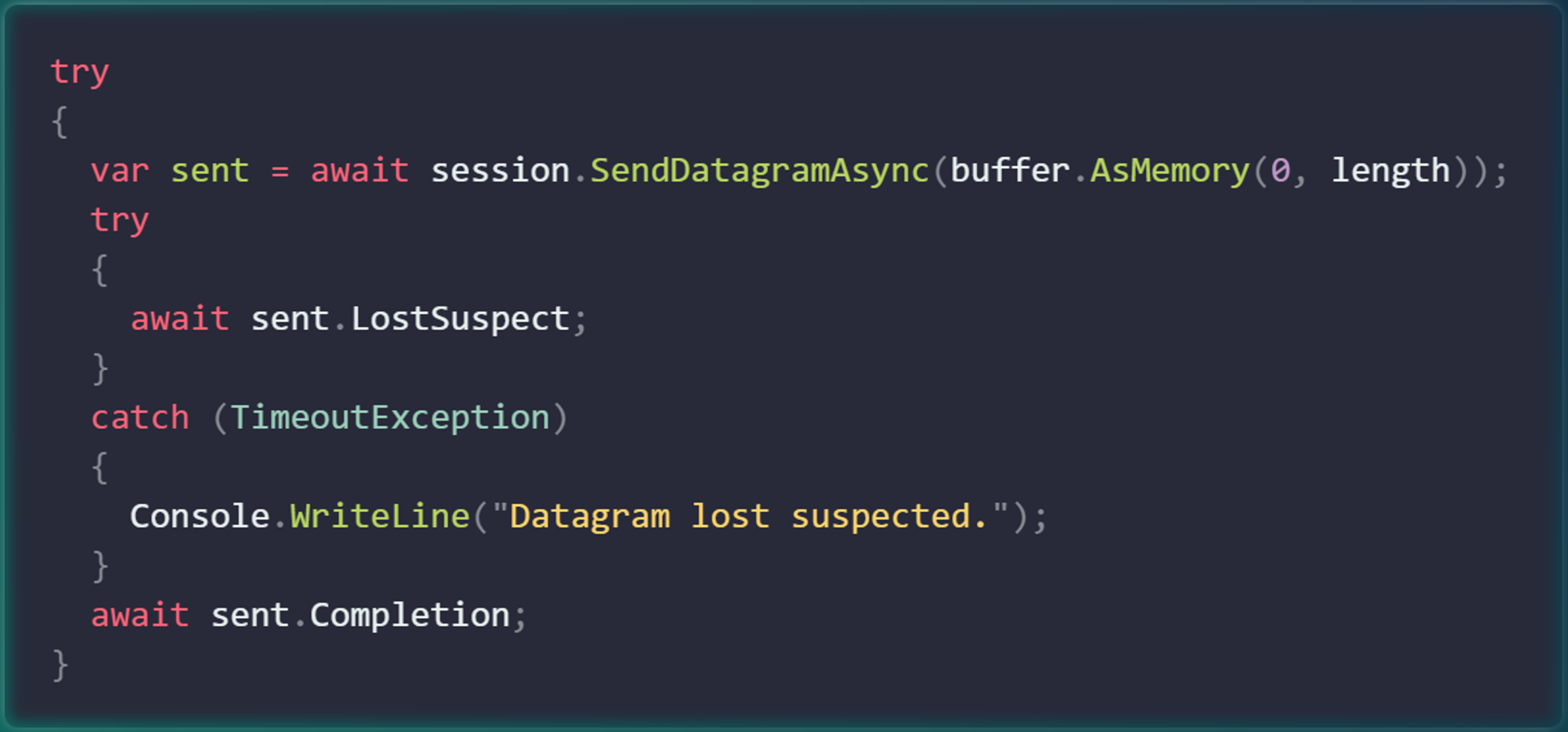
```
static async Task HandleConnectionAsync(WebTransportConnection connection,
 CancellationToken cancellationToken = default)
```

Готовим WebTransport



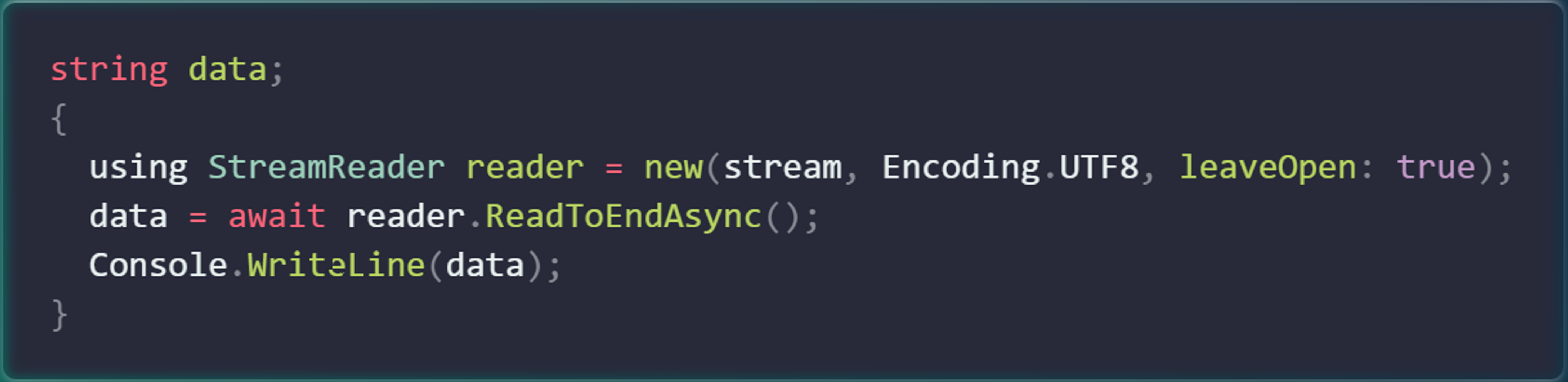
```
session.DatagramReceived += (s, e) =>
{
 var length = e.Length;
 Console.WriteLine($"Получили датаграмму {length} byte(s)");

 var buffer = ArrayPool<byte>.Shared.Rent(length);
 e.CopyTo(buffer);

 string result = string.Create(buffer.Length, buffer, (chars, buf) =>
 {
  for (int i = 0; i < chars.Length; i++) chars[i] = (char)buf[i];
 });
Console.WriteLine($"Получили датаграмму с текстом {result}");
```

Готовим WebTransport



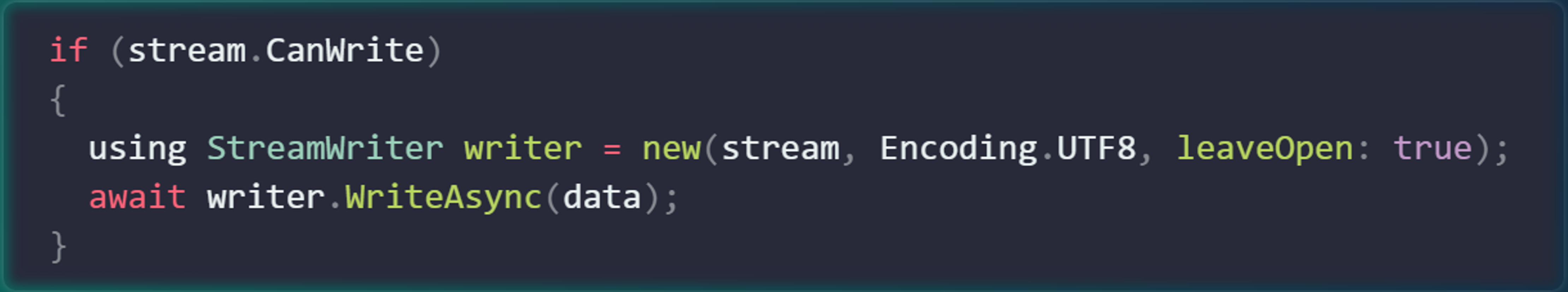
```
try
{
  var sent = await session.SendDatagramAsync(buffer.AsMemory(0, length));
  try
  {
    await sent.LostSuspect;
  }
  catch (TimeoutException)
  {
    Console.WriteLine("Datagram lost suspected.");
  }
  await sent.Completion;
}
```

Готовим WebTransport



```
string data;
{
  using StreamReader reader = new(stream, Encoding.UTF8, leaveOpen: true);
  data = await reader.ReadToEndAsync();
  Console.WriteLine(data);
}
```

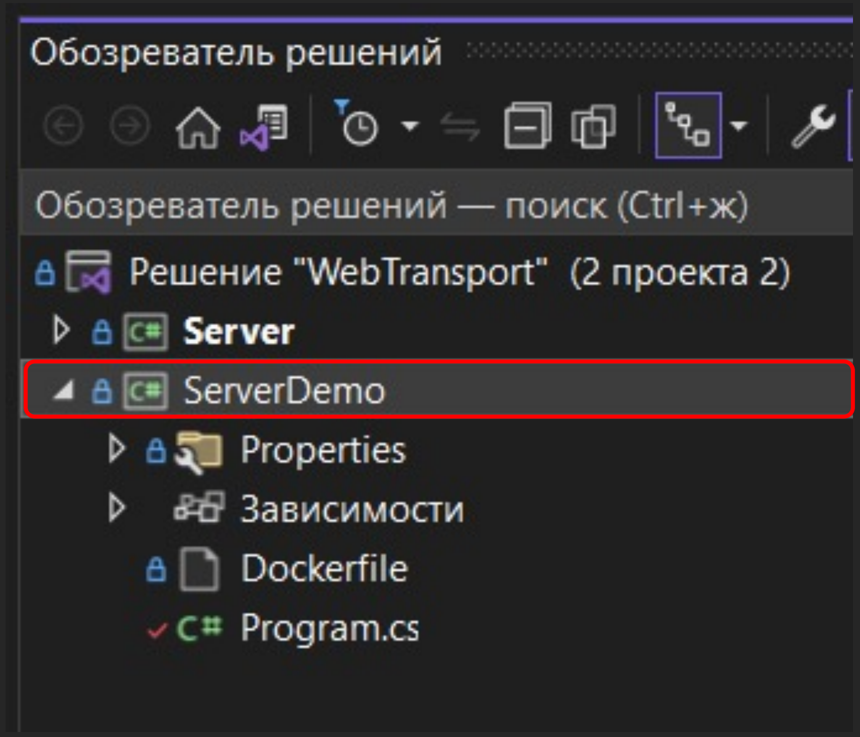
Готовим WebTransport



```
if (stream.CanWrite)
{
  using StreamWriter writer = new(stream, Encoding.UTF8, leaveOpen: true);
  await writer.WriteAsync(data);
}
```

Запускаем!

Запуск WebTransport



Обозреватель решений

Обозреватель решений — поиск (Ctrl+ж)

Решение "WebTransport" (2 проекта 2)

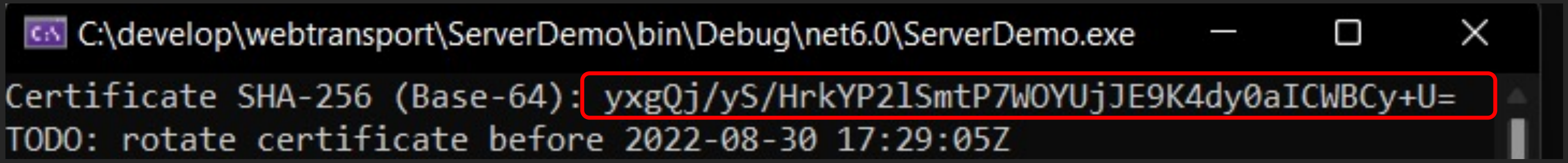
- Server
- ServerDemo
  - Properties
  - Зависимости
  - Dockerfile
  - Program.cs

Запуск WebTransport

1) Открываем терминал в папке ServerDemo

2) Выполняем команду dotnet run

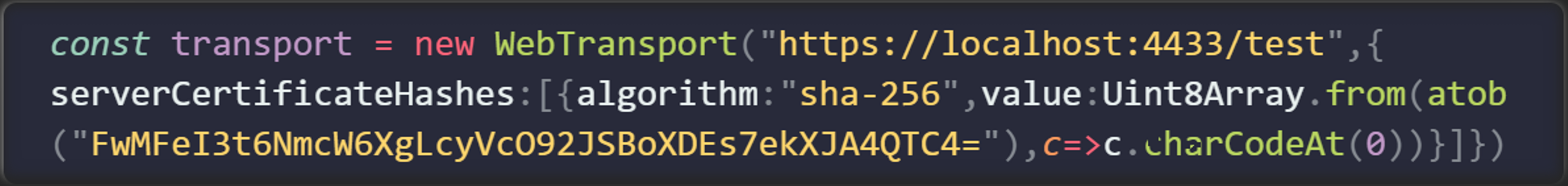
Запуск WebTransport



C:\develop\webtransport\ServerDemo\bin\Debug\net6.0\ServerDemo.exe

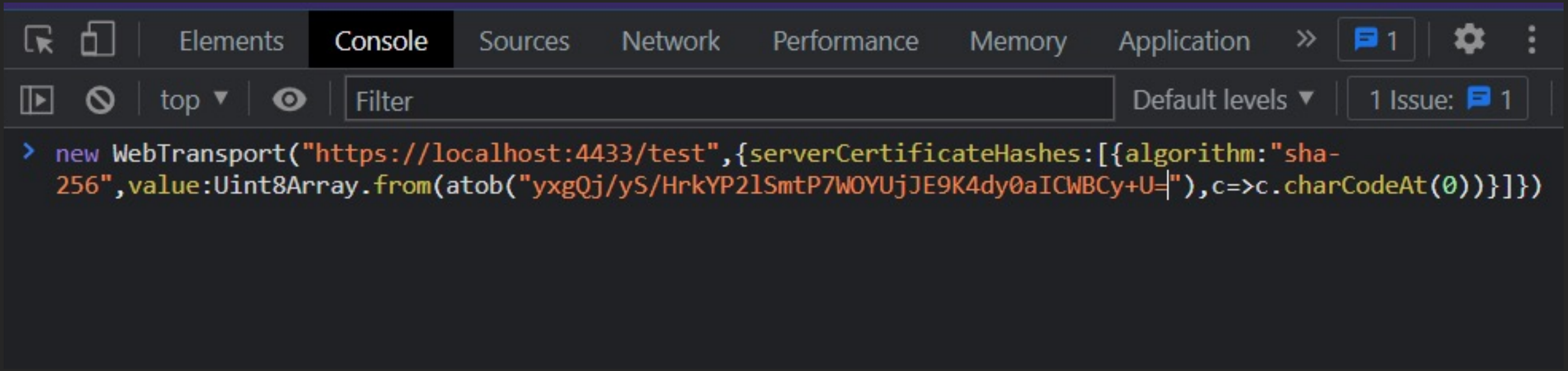
```
Certificate SHA-256 (Base-64): yxgQj/yS/HrkYP2lSmtP7WOYUjJE9K4dy0aICWBCy+U=
TODO: rotate certificate before 2022-08-30 17:29:05Z
```

Запуск WebTransport



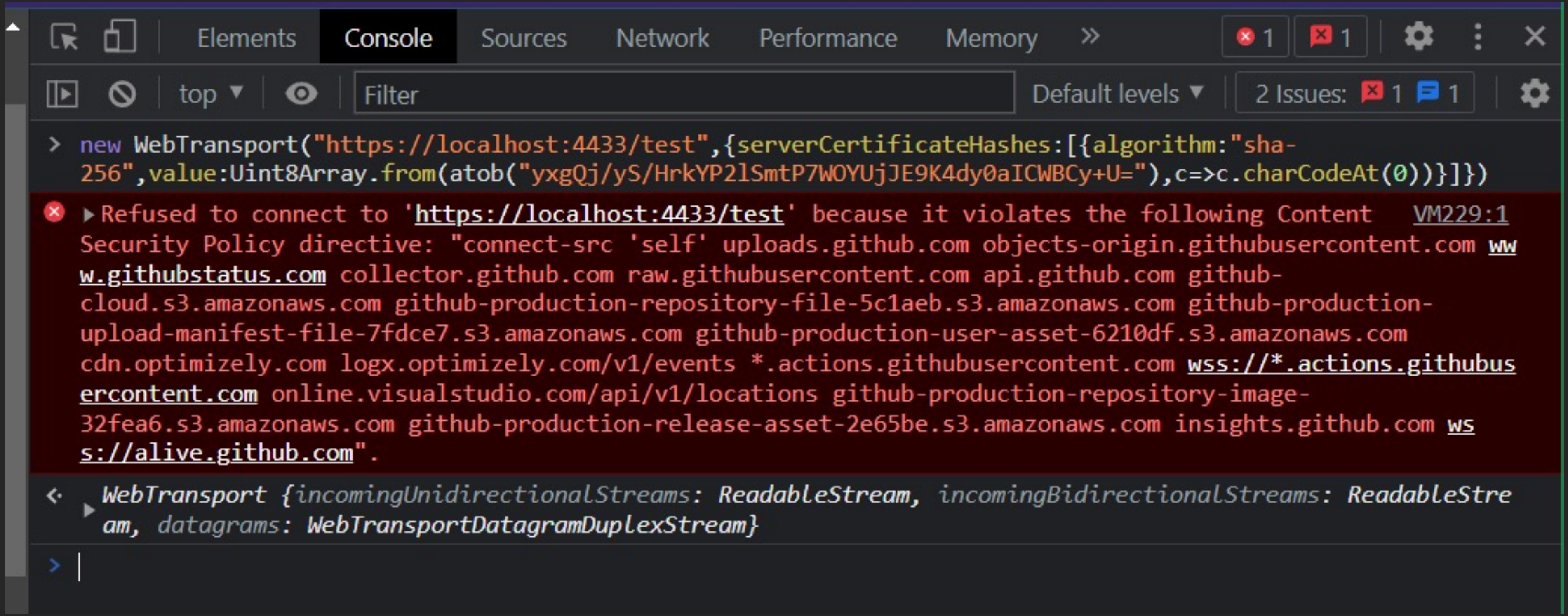
```
const transport = new WebTransport("https://localhost:4433/test",{
serverCertificateHashes:[{algorithm:"sha-256",value:Uint8Array.from(atob
("FwMFeI3t6NmcW6XgLcyVcO92JSBoXDEs7ekXJA4QTC4="),c=>c.charCodeAt(0))}]})
```

Запуск WebTransport



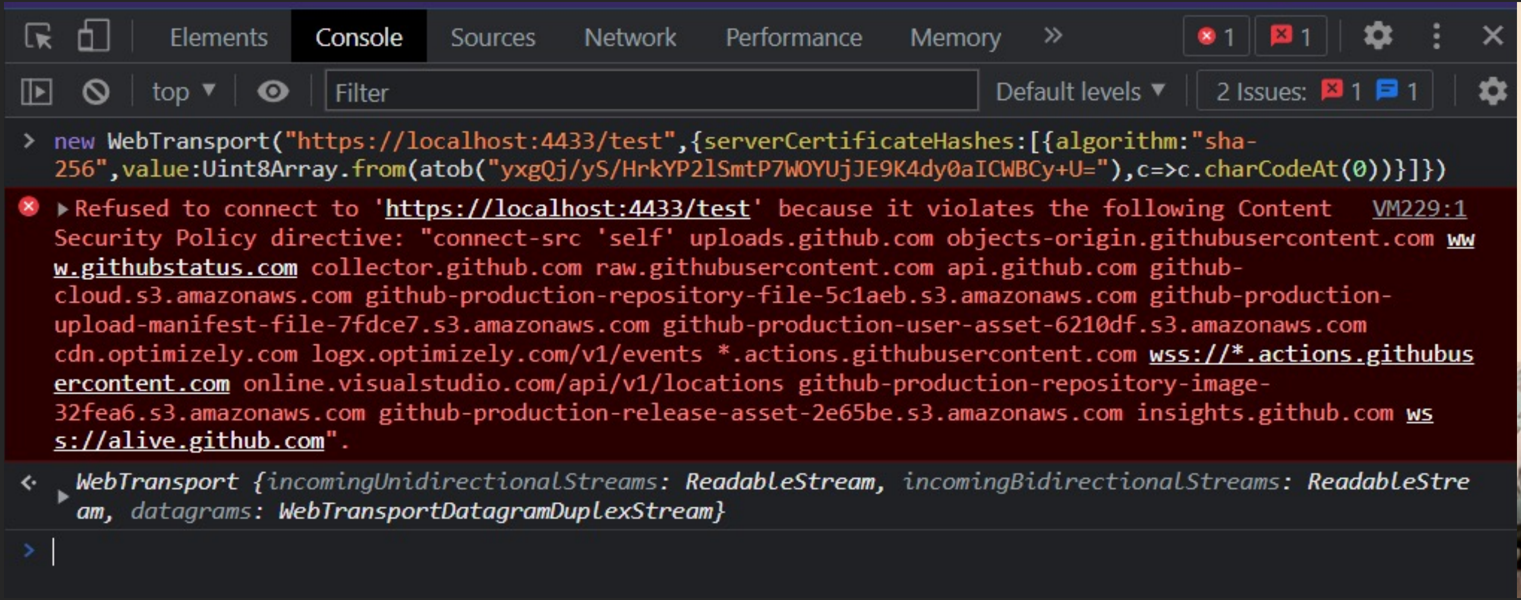
```
> new WebTransport("https://localhost:4433/test",{serverCertificateHashes:[{algorithm:"sha-
  256",value:Uint8Array.from(atob("yxgQj/yS/HrkYP2lSmtP7WOYUjJE9K4dy0aICWBCy+U="),c=>c.charCodeAt(0))}]})
```

Запуск WebTransport



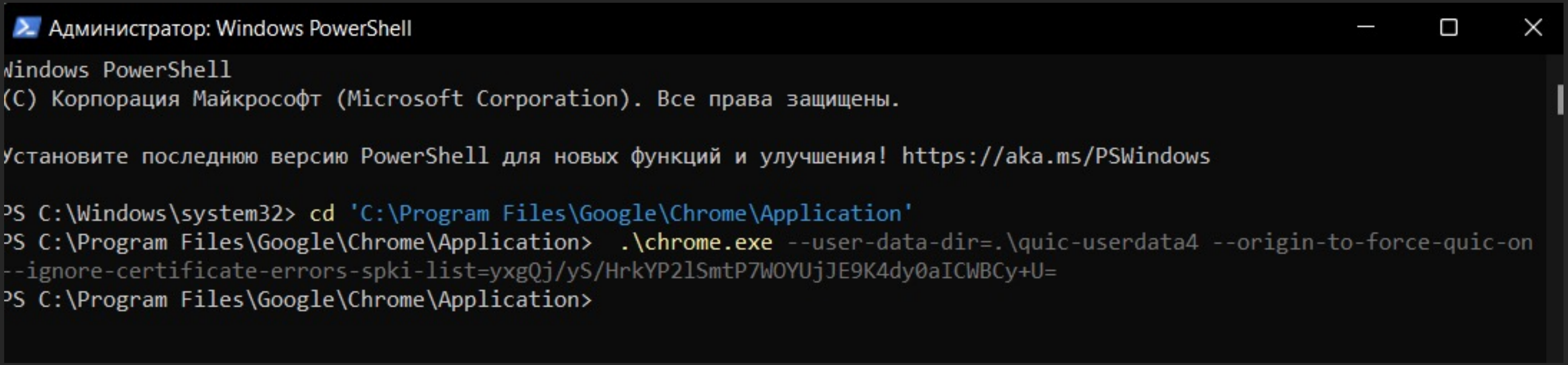
```
> new WebTransport("https://localhost:4433/test",{serverCertificateHashes:[{algorithm:"sha-
  256",value:Uint8Array.from(atob("yxgQj/yS/HrkYP2lSmtP7WOYUjJE9K4dy0aICWBCy+U="),c=>c.charCodeAt(0))}]})
✖ ▸Refused to connect to 'https://localhost:4433/test' because it violates the following Content    VM229:1
  Security Policy directive: "connect-src 'self' uploads.github.com objects-origin.githubusercontent.com ww
  w.githubstatus.com collector.github.com raw.githubusercontent.com api.github.com github-
  cloud.s3.amazonaws.com github-production-repository-file-5c1aeb.s3.amazonaws.com github-production-
  upload-manifest-file-7fdce7.s3.amazonaws.com github-production-user-asset-6210df.s3.amazonaws.com
  cdn.optimizely.com logx.optimizely.com/v1/events *.actions.githubusercontent.com wss://*.actions.githubus
  ercontent.com online.visualstudio.com/api/v1/locations github-production-repository-image-
  32fea6.s3.amazonaws.com github-production-release-asset-2e65be.s3.amazonaws.com insights.github.com ws
  s://alive.github.com".
< ▸ WebTransport {incomingUnidirectionalStreams: ReadableStream, incomingBidirectionalStreams: ReadableStre
    am, datagrams: WebTransportDatagramDuplexStream}
> |
```

Запуск WebTransport



```
> new WebTransport("https://localhost:4433/test",{serverCertificateHashes:[{algorithm:"sha-
  256",value:Uint8Array.from(atob("yxgQj/yS/HrkYP2lSmtP7WOYUjJE9K4dy0aICWBCy+U="),c=>c.charCodeAt(0))}]})
✕ ▸Refused to connect to 'https://localhost:4433/test' because it violates the following Content   VM229:1
  Security Policy directive: "connect-src 'self' uploads.github.com objects-origin.githubusercontent.com ww
  w.githubstatus.com collector.github.com raw.githubusercontent.com api.github.com github-
  cloud.s3.amazonaws.com github-production-repository-file-5c1aeb.s3.amazonaws.com github-production-
  upload-manifest-file-7fdce7.s3.amazonaws.com github-production-user-asset-6210df.s3.amazonaws.com
  cdn.optimizely.com logx.optimizely.com/v1/events *.actions.githubusercontent.com wss://*.actions.githubus
  ercontent.com online.visualstudio.com/api/v1/locations github-production-repository-image-
  32fea6.s3.amazonaws.com github-production-release-asset-2e65be.s3.amazonaws.com insights.github.com ws
  s://alive.github.com".
<· ▸WebTransport {incomingUnidirectionalStreams: ReadableStream, incomingBidirectionalStreams: ReadableStre
  am, datagrams: WebTransportDatagramDuplexStream}
> |
```

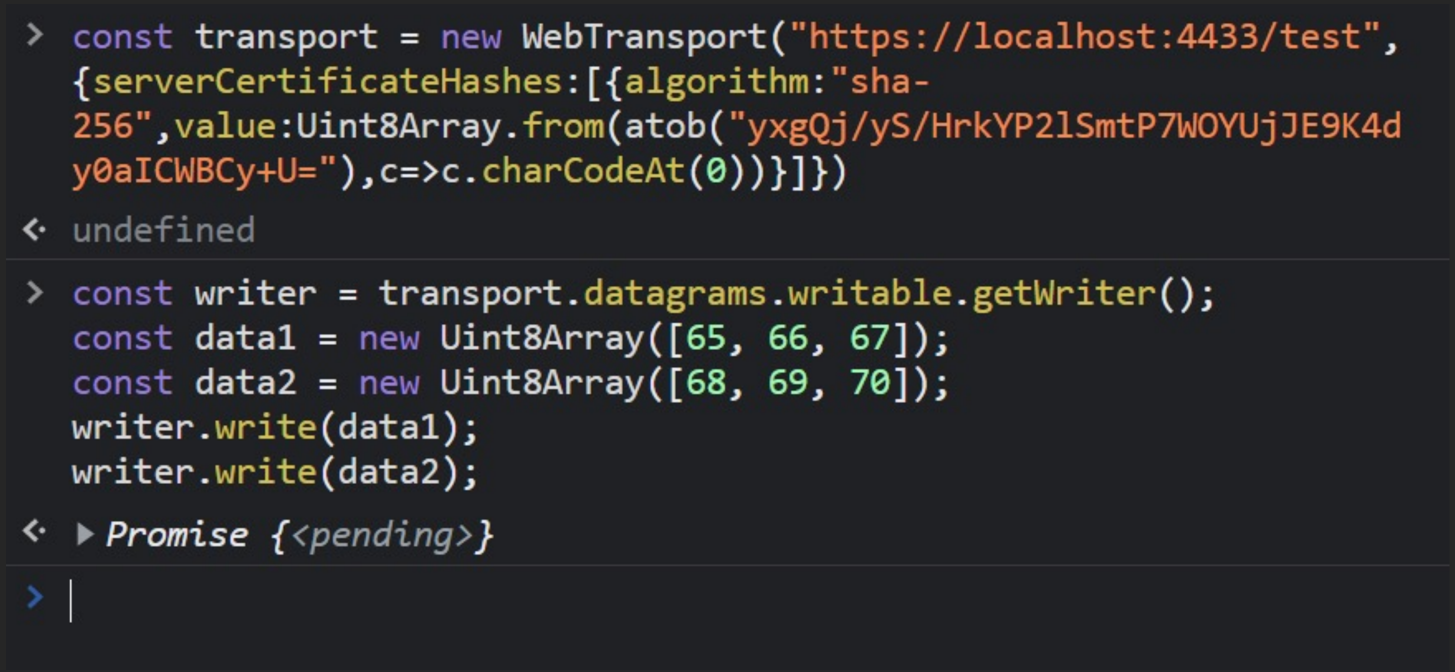
Запуск Chrome



Администратор: Windows PowerShell

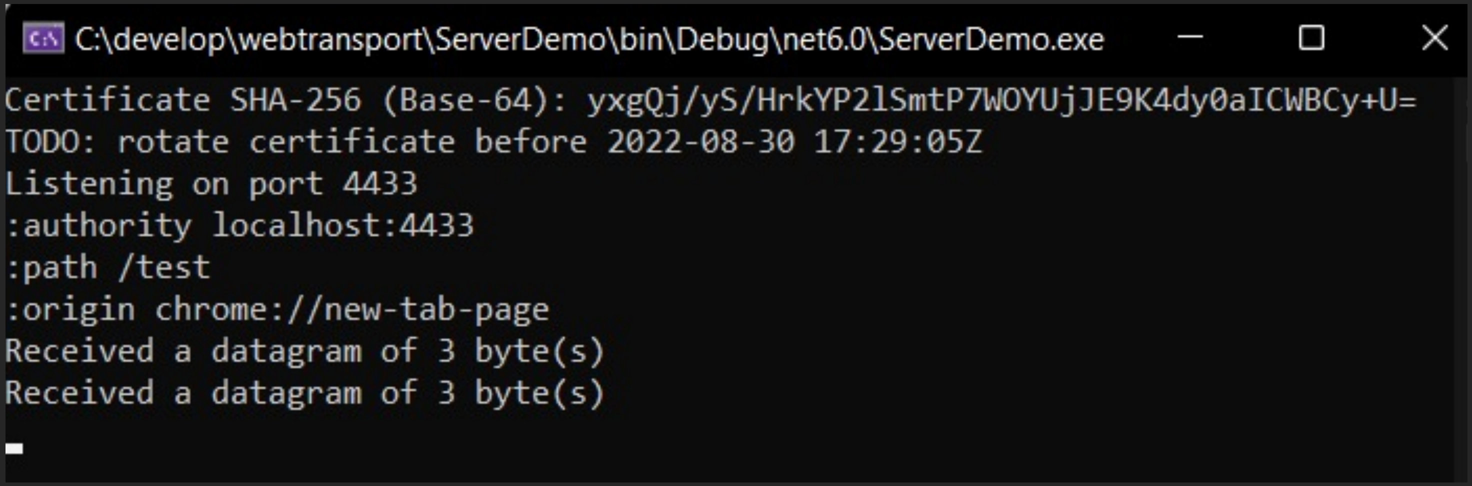
```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Windows\system32> cd 'C:\Program Files\Google\Chrome\Application'
PS C:\Program Files\Google\Chrome\Application>  .\chrome.exe --user-data-dir=.\quic-userdata4 --origin-to-force-quic-on
--ignore-certificate-errors-spki-list=yxgQj/yS/HrkYP2lSmtP7WOYUjJE9K4dy0aICWBCy+U=
PS C:\Program Files\Google\Chrome\Application>
```

Запуск WebTransport



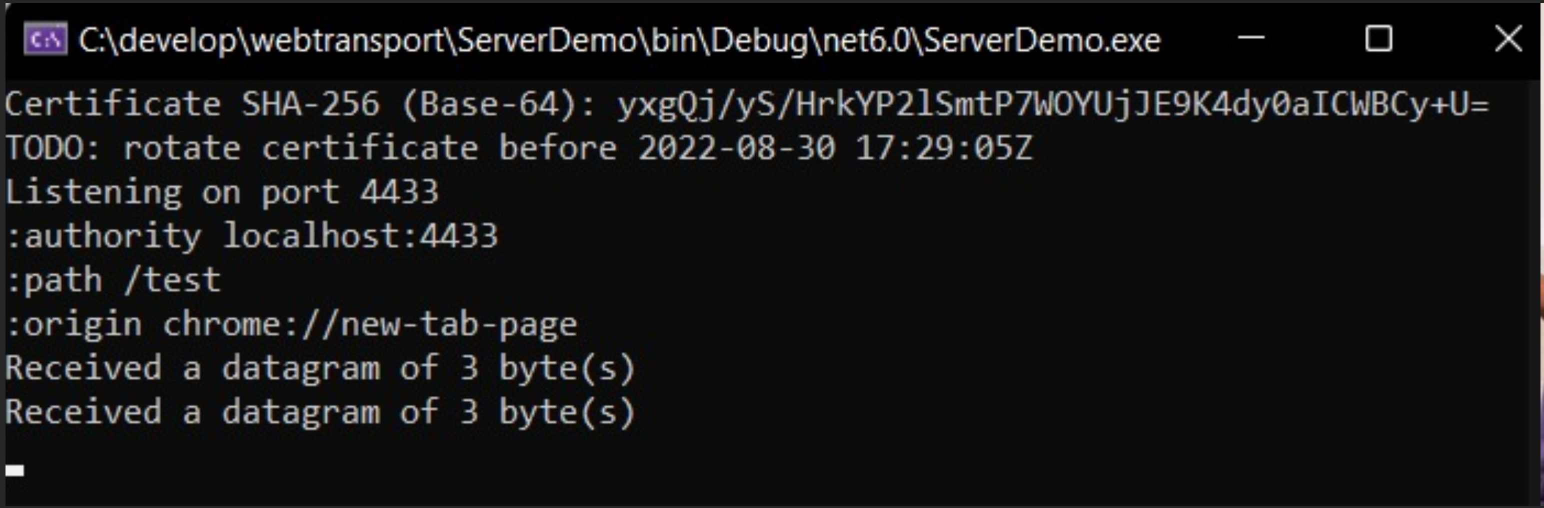
```
> const transport = new WebTransport("https://localhost:4433/test",
  {serverCertificateHashes:[{algorithm:"sha-
  256",value:Uint8Array.from(atob("yxgQj/yS/HrkYP2lSmtP7WOYUjJE9K4d
  y0aICWBCy+U="),c=>c.charCodeAt(0))}]})
<· undefined
> const writer = transport.datagrams.writable.getWriter();
  const data1 = new Uint8Array([65, 66, 67]);
  const data2 = new Uint8Array([68, 69, 70]);
  writer.write(data1);
  writer.write(data2);
<· ▶ Promise {<pending>}
> |
```

Запуск WebTransport


C:\develop\webtransport\ServerDemo\bin\Debug\net6.0\ServerDemo.exe

```
Certificate SHA-256 (Base-64): yxgQj/yS/HrkYP2lSmtP7WOYUjJE9K4dy0aICWBCy+U=
TODO: rotate certificate before 2022-08-30 17:29:05Z
Listening on port 4433
:authority localhost:4433
:path /test
:origin chrome://new-tab-page
Received a datagram of 3 byte(s)
Received a datagram of 3 byte(s)
```

Запуск WebTransport

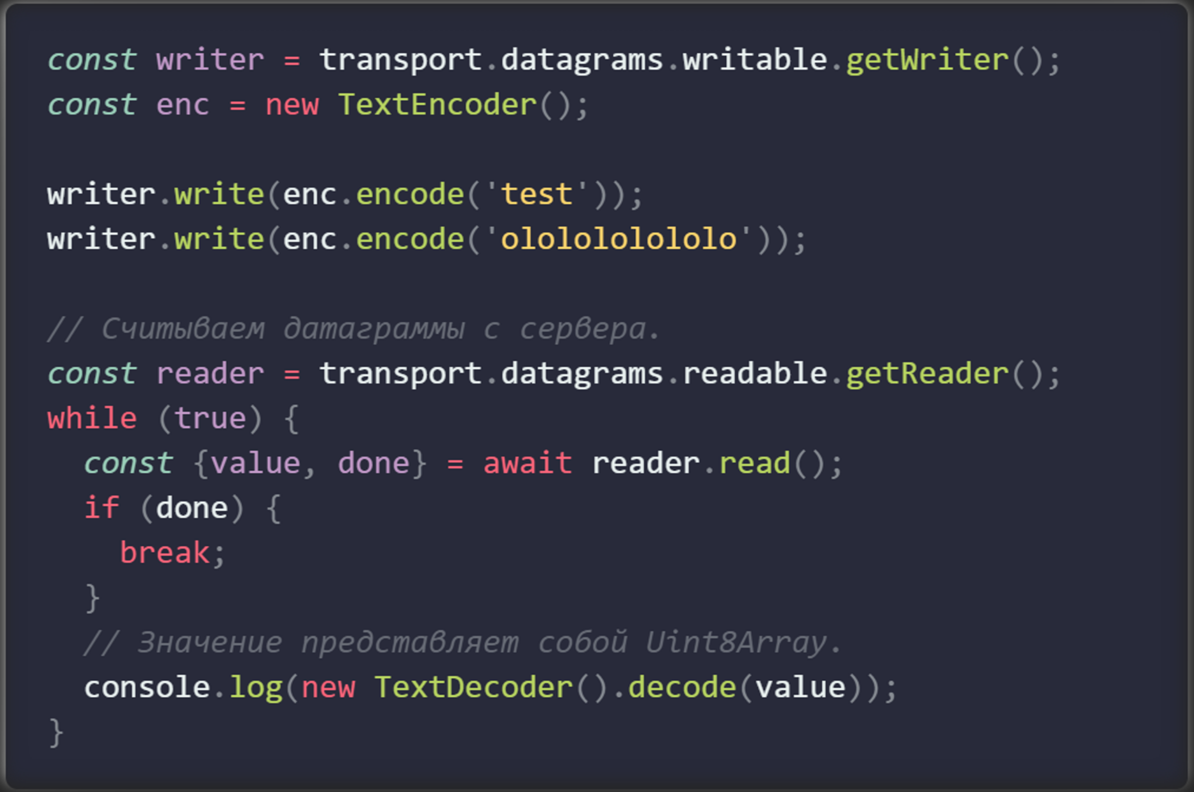


C:\develop\webtransport\ServerDemo\bin\Debug\net6.0\ServerDemo.exe

```
Certificate SHA-256 (Base-64): yxgQj/yS/HrkYP2lSmtP7WOYUjJE9K4dy0aICWBCy+U=
TODO: rotate certificate before 2022-08-30 17:29:05Z
Listening on port 4433
:authority localhost:4433
:path /test
:origin chrome://new-tab-page
Received a datagram of 3 byte(s)
Received a datagram of 3 byte(s)
```

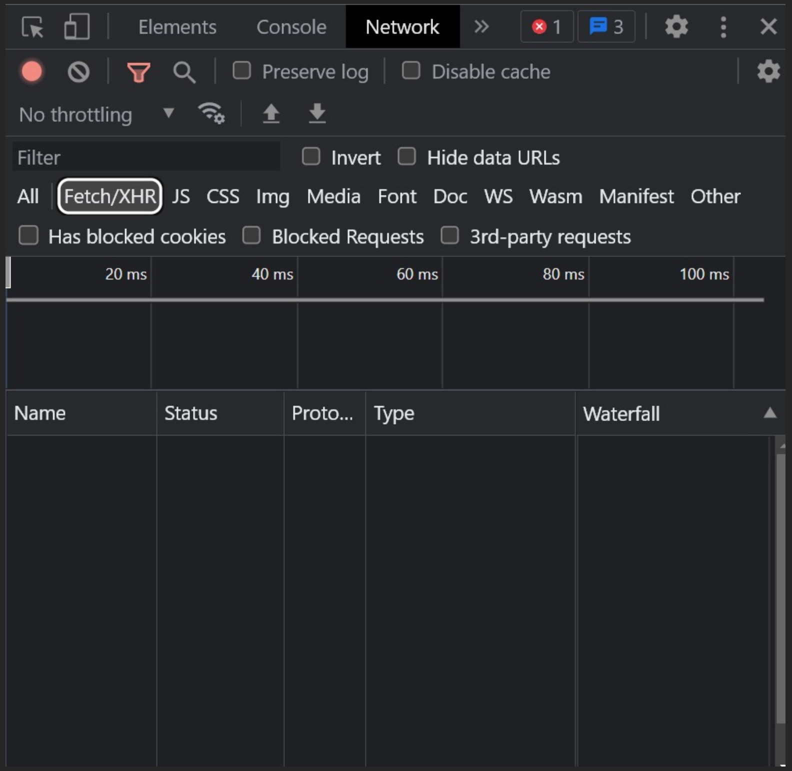
Запуск WebTransport

Пример с отправкой текста

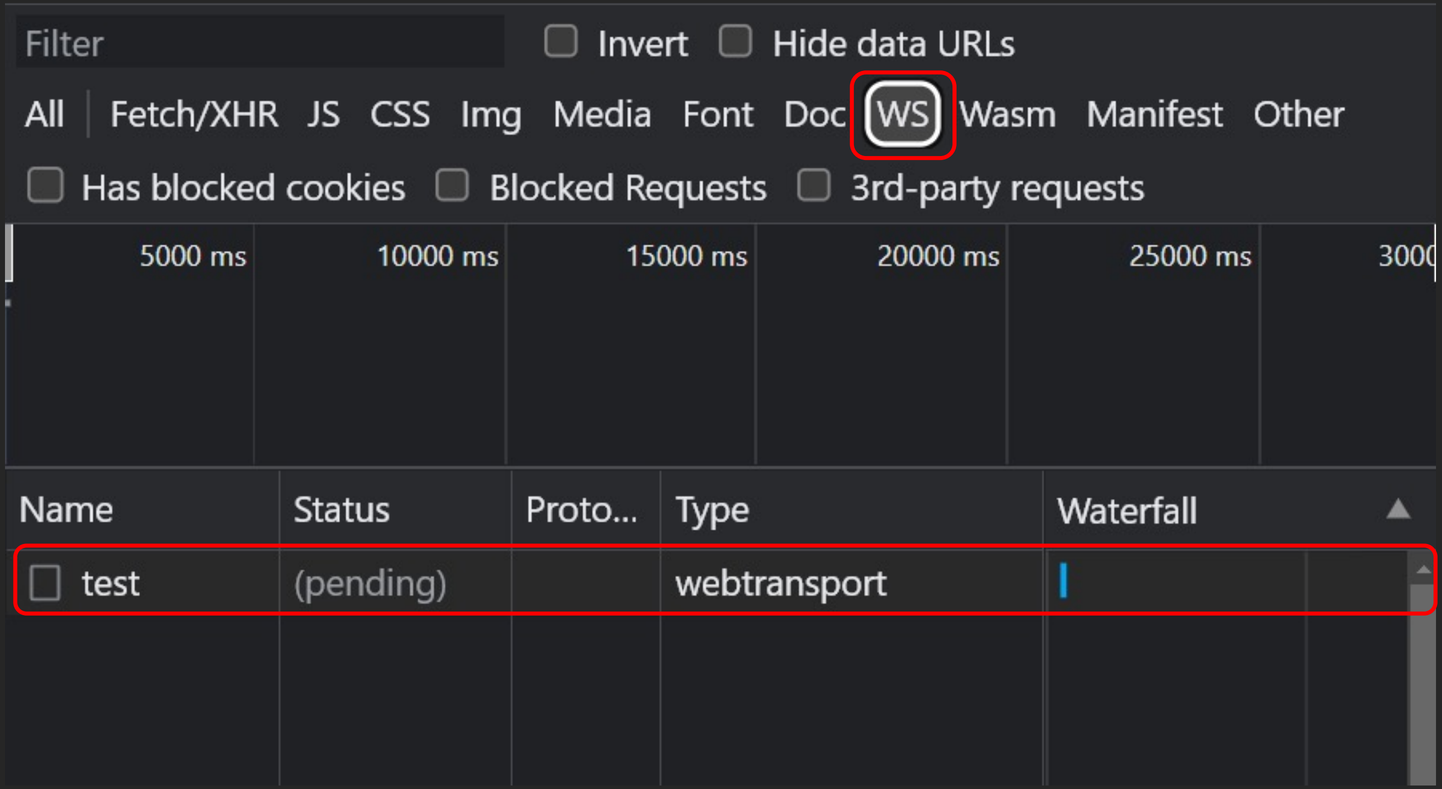


```
const writer = transport.datagrams.writable.getWriter();
const enc = new TextEncoder();

writer.write(enc.encode('test'));
writer.write(enc.encode('ololololololo'));

// Считываем датаграммы с сервера.
const reader = transport.datagrams.readable.getReader();
while (true) {
  const {value, done} = await reader.read();
  if (done) {
    break;
  }
  // Значение представляет собой Uint8Array.
  console.log(new TextDecoder().decode(value));
}
```

Запуск WebTransport



Запуск WebTransport



Filter □ Invert □ Hide data URLs

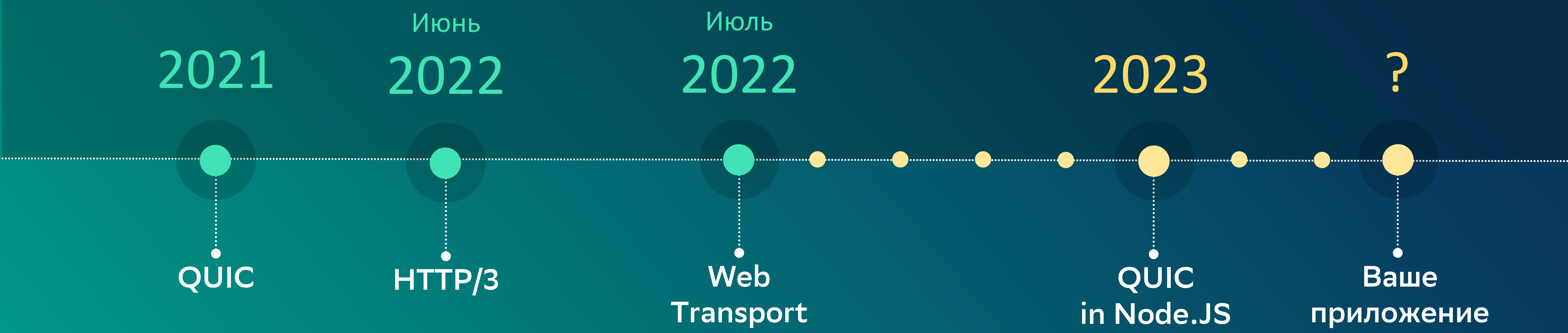
All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other

□ Has blocked cookies □ Blocked Requests □ 3rd-party requests

Name	Status	Proto...	Type	Waterfall
□ test	(pending)		webtransport	

Новые технологии на подходе

Давайте будем к ним готовы



2021

QUIC

Июнь

2022

HTTP/3

Июль

2022

Web Transport

2023

QUIC in Node.JS

?

Ваше приложение

Вопросы?

Андрей Власов

tg: @InnoAndrez
E-mail: innoavvlasov@gmail.com

Василий Маркитан

tg: @Markitan_V
E-mail: mkt.dev@yandex.ru