

Your network connection is slow. Выявляем проблемы с сетью у пользователя

Павел Дадыкин

ROGII



Frontend
Conf
2025

Обо мне

Павел Дадыкин

- 📍 Живу в Ульяновске
- 💻 10 лет во фронтенде
- 🔥 6 лет в компании ROGII
- 💻 Выступал на FrontendConf 2024





-  Разработка продуктов для нефтегазовой индустрии
-  Geoscience-решения
-  Много математических расчётов
-  Desktop, Web, iOS, Android, SDK, Public API

NUSmoke1 > Lateral1

Save View

Following

Rewind



NU

Object Tree

Search

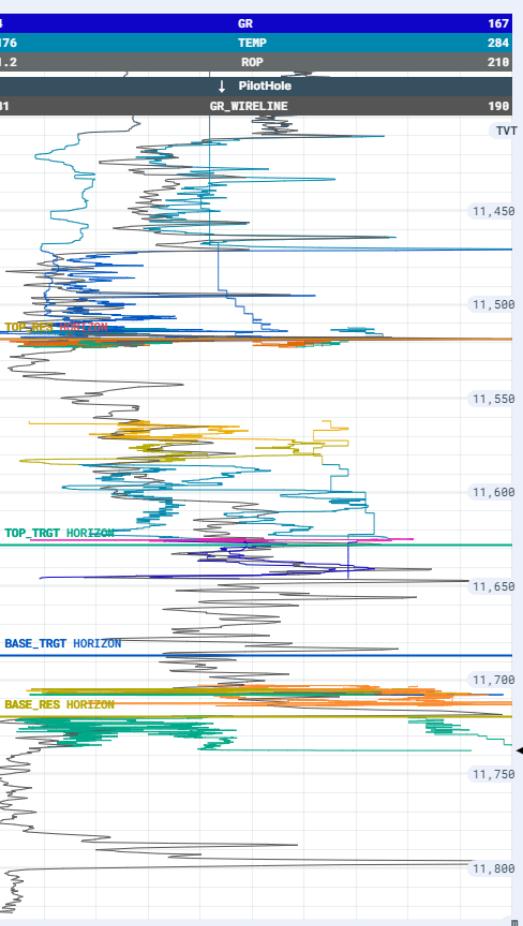
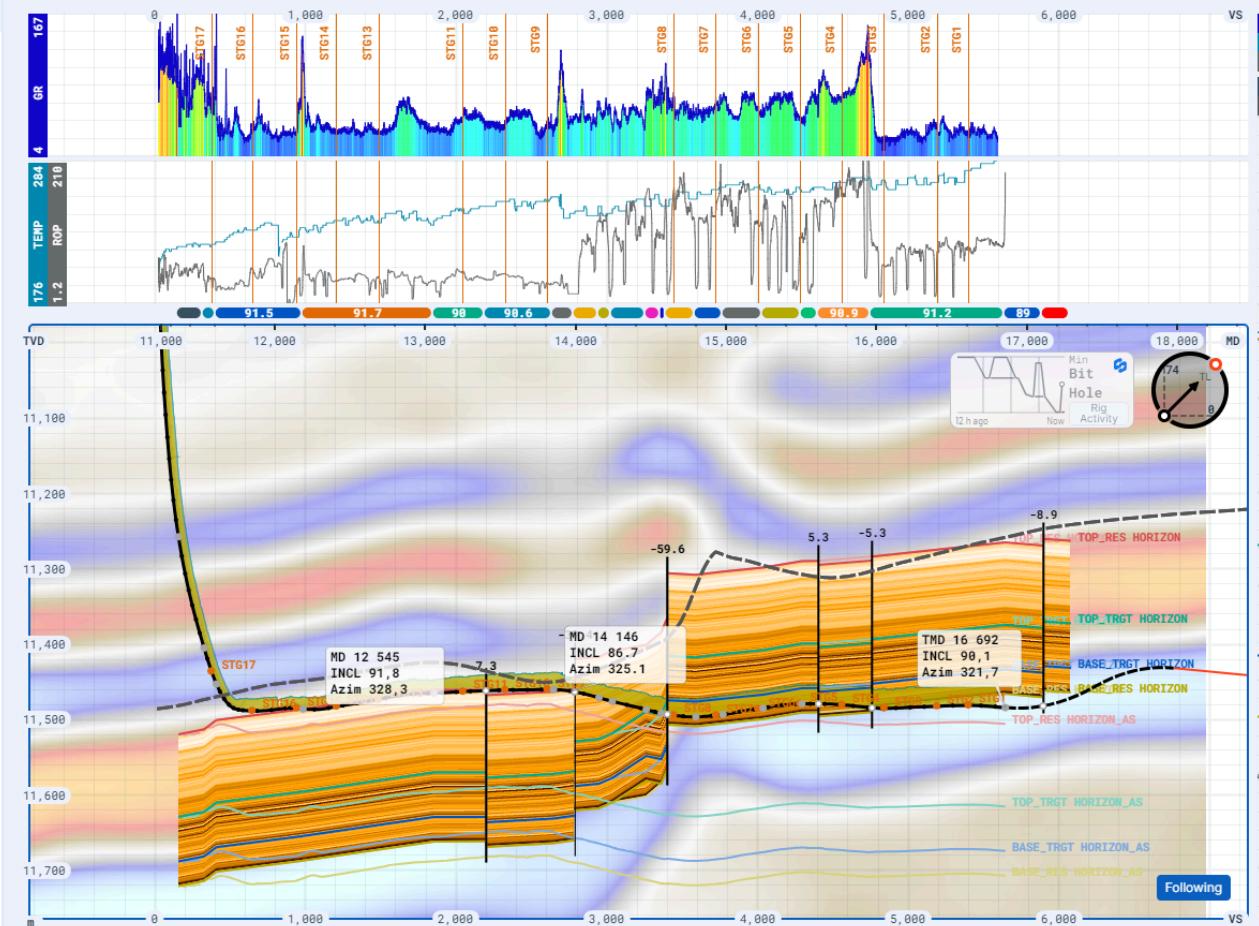
Wells

- ✓ Lateral1
 - > Interpretation2
 - > Interpretation1
 - > Interpretation3
 - > AS_Interpretation1
 - ✓ ROP
 - ✓ TEMP
 - ✓ GR

- + Topset1
 - ✓ STG17
 - ✓ STG16
 - ✓ STG15
 - ✓ STG14
 - ✓ STG13
 - ✓ STG11
 - ✓ STG10
 - ✓ STG9
 - ✓ STG8
 - ✓ STG7
 - ✓ STG6
 - ✓ STG5
 - ✓ STG4
 - ✓ STG3
 - ✓ STG2
 - ✓ STG1
- ✓ seis_bright.jpg
- ✓ Target Line1
- ✓ Comments1

Typewells

- ✓ PilotHole
 - ✓ GR_wireline
- + Topset1
- ✓ Top_Res
- ✓ Top_Trgt
- ✓ Base_Trgt
- ✓ Base_Res



VS: 5,976

MD: 17,193.7

INCL: 93.22

AZIM GN: 324.6

DLS: 0.4

TVD: 11,476.1

TVDSS: -11,082.1

TVT:

N/S: 4,891.98

E/W: -3,432.61

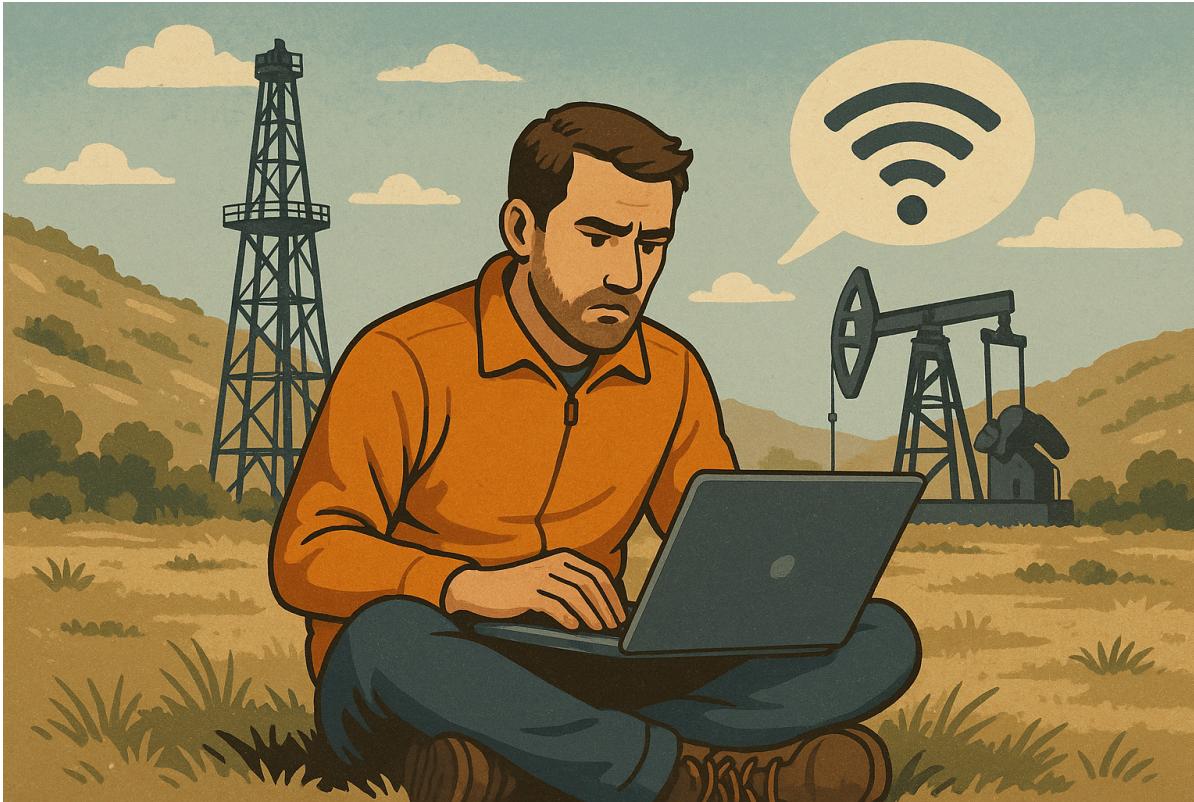
X: 496,567.4

Y: 604,892.0

Работа из офиса



Работа в поле



О чём доклад

- ? Зачем пользователям знать о проблемах с соединением
- ⚙ Как наш сервис должен реагировать
- 🌐 Какие браузерные API работают, а какие нет
 - ⚡ navigator.onLine
 - 🛰 Network Information API
 - 📊 Resource Timing API

Зачем пользователям знать о проблемах с их соединением?

Главная цель – улучшить UX



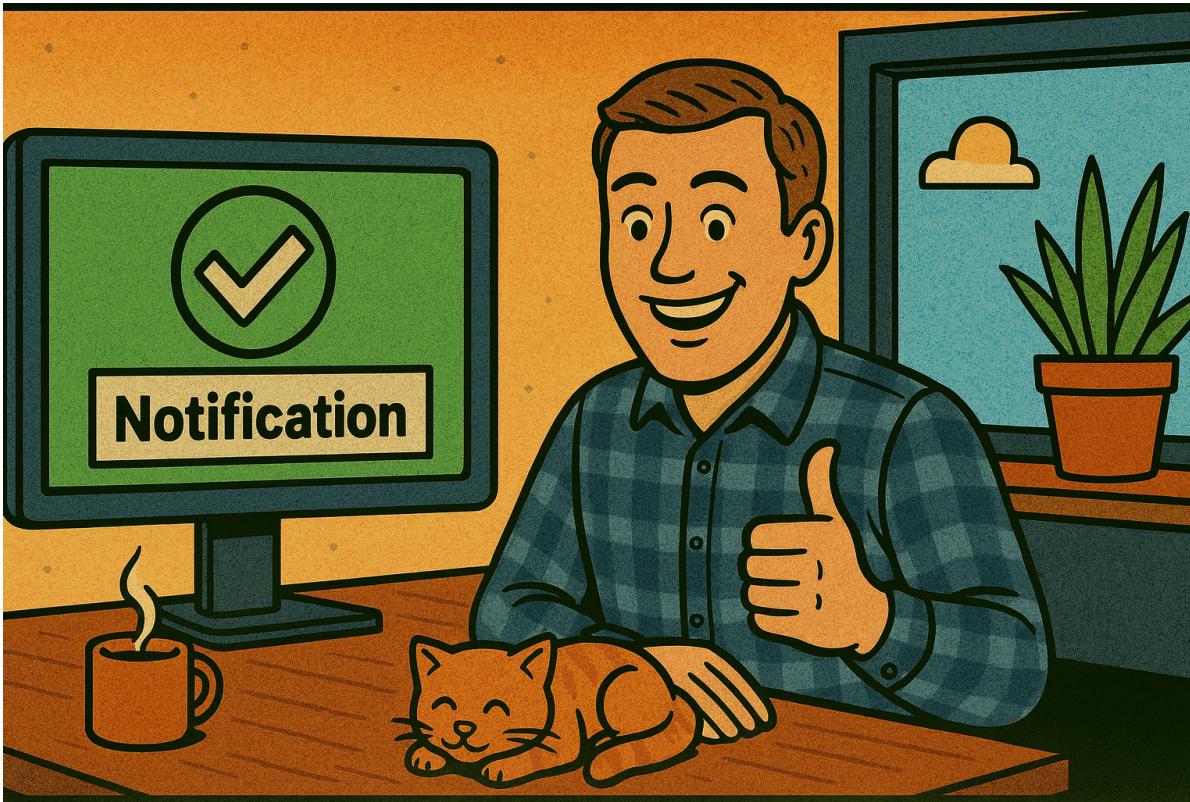
Пользователь не всегда понимает, что происходит



Он может ждать live-данные, а они не придут



Уведомление повышает доверие к продукту



Типы проблем

 **Полное отключение** от интернета

 Временные **скачки скорости**

 Постоянно **низкая пропускная способность**

**Браузер должен уметь всё это
определять**

Определяем Online/Offline

```
window.navigator.onLine
```



```
if (navigator.onLine) {  
    console.log("online");  
} else {  
    console.log("offline");  
}
```

События navigator.onLine

```
window.addEventListener('online', () => {/* сеть есть */});
```

```
window.addEventListener('offline', () => {/* сети нет */});
```



Особенности navigator.onLine

- Показывает только есть ли **подключение к сети**
- **Не делает** пинг, запрос или проверку доступности
- Нет информации о скорости или стабильности

navigator.onLine === true

- 😊 С соединением всё в порядке
- 😢 DNS не работает
- 😢 Wi-Fi есть, но интернет «упал»
- 😢 Wi-Fi требует авторизации
- 😢 Проблемы с VPN

Что же показывает navigator.onLine?

false → Проблемы с сетью точно есть 😞

true → **Возможно**, с сетью всё в порядке 😐

Не стоит полагаться только на этот способ!

Как же определить скорость соединения?

Network Information API

Браузерное API для получения информации о типе подключения и скорости

`window.navigator.connection`

Свойства navigator.connection

Свойство	Описание
effectiveType	Тип соединения: 'slow-2g', '2g', '3g', '4g'
type	Тип сети: 'wifi', 'cellular', 'bluetooth', 'ethernet', ...
downlink	Оценка скорости в Mbps
downlinkMax	Максимально возможная скорость
rtt	Round-trip time (RTT) в миллисекундах
saveData	Включён ли режим экономии трафика

Network Information API



Limited availability



Network Information API Sample

Live Output

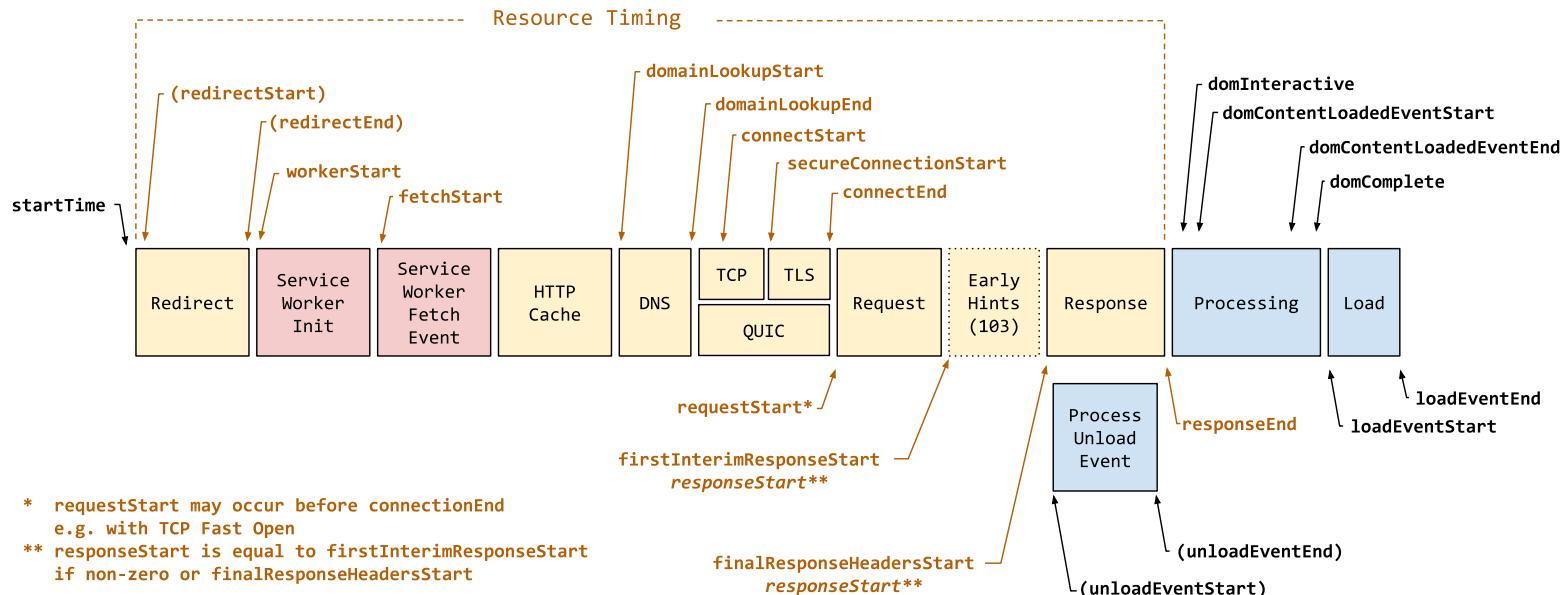
```
    type: undefined
  downlink: 10 Mb/s
    rtt: 100 ms
  downlinkMax: undefined Mb/s
effectiveType: 4g
  saveData: false
```



Resource Timing API

- API для анализа времени загрузки ресурсов на странице
- `performance.getEntriesByType('resource')`
- Позволяет отслеживать **фактическую производительность** на клиенте
- Работает с: `img`, `script`, `fetch`, `link`, `xhr` и др.
- Очень гибкий инструмент

Resource Timing API



Пример Performance Resource Timing

```
▼ 13: PerformanceResourceTiming
  connectEnd: 436.0999999962747
  connectStart: 436.0999999962747
  decodedBodySize: 541
  deliveryType: ""
  domainLookupEnd: 436.0999999962747
  domainLookupStart: 436.0999999962747
  duration: 325.90000000037253
  encodedBodySize: 541
  entryType: "resource"
  fetchStart: 436.0999999962747
  finalResponseHeadersStart: 756.399999999069
  firstInterimResponseStart: 0
  initiatorType: "css"
  name: "https://developer.mozilla.org/static/media/bluesky.ced6c28203d89d3910ab.svg"
  nextHopProtocol: "h2"
  redirectEnd: 0
  redirectStart: 0
  renderBlockingStatus: "non-blocking"
  requestStart: 685.899999999069
  responseEnd: 762
  responseStart: 756.399999999069
  responseStatus: 200
  secureConnectionStart: 436.0999999962747
  ▶ serverTiming: []
  startTime: 436.0999999962747
  transferSize: 841
  workerStart: 0
```

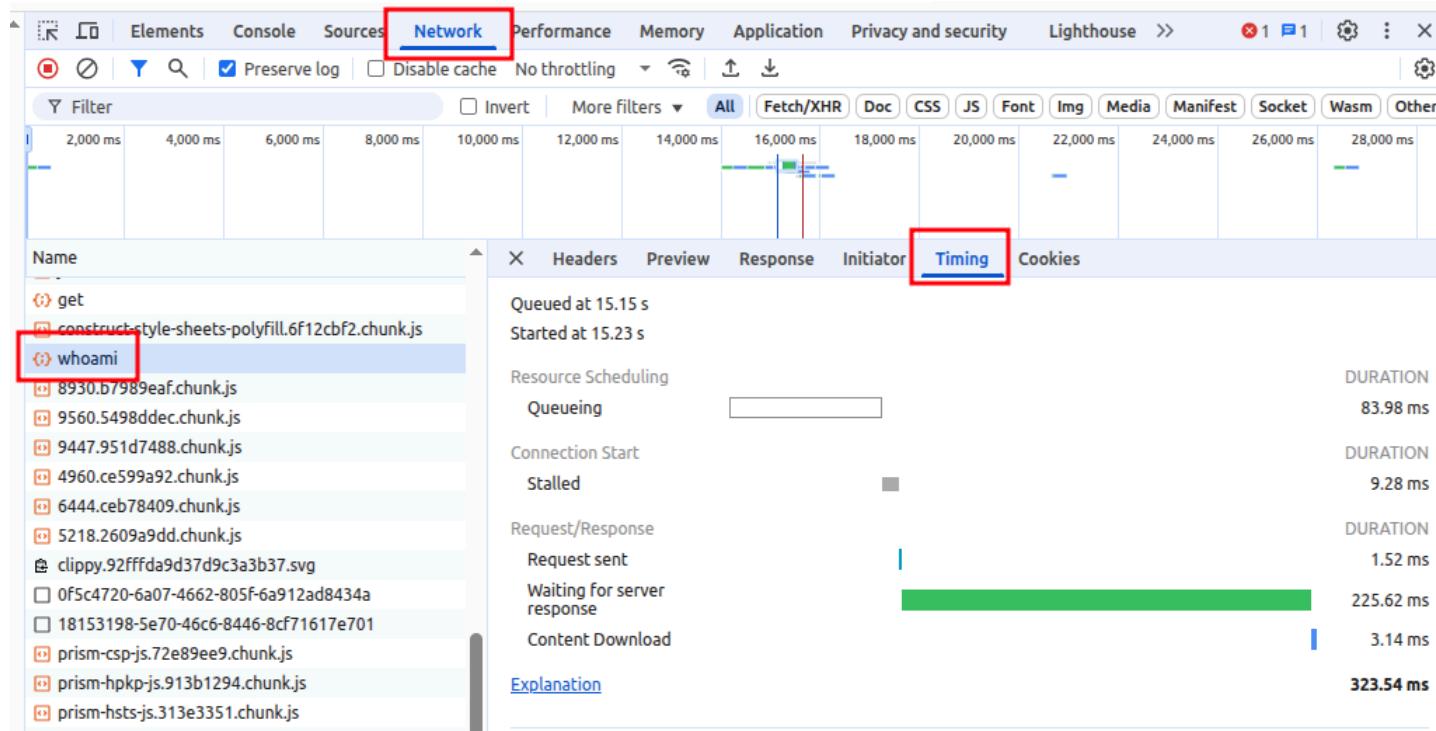
Типичные метрики Resource Timing

Метрика	Формула	Назначение
TCP Handshake	<code>connectEnd - connectStart</code>	Время установления TCP-соединения
DNS Lookup	<code>domainLookupEnd - domainLookupStart</code>	Время разрешения DNS-имени
Redirect Time	<code>redirectEnd - redirectStart</code>	Задержка из-за переадресации
Request Time	<code>responseStart - requestStart</code>	Время между отправкой запроса и первым байтом ответа
TLS Negotiation	<code>requestStart - secureConnectionStart</code>	Время на установление HTTPS-соединения
Fetch Time	<code>responseEnd - fetchStart</code>	Общее время загрузки (без редиректов)

Типичные метрики Resource Timing

Метрика	Формула	Назначение
Service Worker	<code>fetchStart - workerStart</code>	Задержка на обработку запроса в Service Worker
Контент сжат?	<code>decodedBodySize ≠ encodedBodySize</code>	Проверка наличия сжатия (например, gzip)
Использован кэш?	<code>transferSize === 0</code>	Загрузка из локального кэша
Современный протокол?	<code>nextHopProtocol</code>	Ожидается: <code>h2</code> или <code>h3</code> (HTTP/2/3)
Блокирует отрисовку?	<code>renderBlockingStatus</code>	Указывает, задерживает ли ресурс first paint

Timing w/ Developer Tools



Сценарии использования Resource Timing API

-  Анализ реальной производительности на клиенте
-  Выявление узких мест в производительности
-  Улучшение логики загрузки и отложенной инициализации
-  Профилирование динамически загружаемых ресурсов



Сбор пользовательских метрик

Можно записывать в аналитику:

-  Время загрузки изображений, скриптов, иконок, шрифтов
-  Как влияет скорость соединения на время отображения UI
-  Географию/сети пользователей с медленными загрузками

Определяем медленное соединение

1. Следим за **fetch** запросами
2. Смотрим **время** загрузки (duration)
3. Смотрим **объём** (transferSize, **в байтах**)
4. Вычисляем **скорость** (transferSize / duration)
5. Если скорость **ниже** какого-то порога - сигнализируем

Получение информации о запросах

Используем Performance Observer

```
const observer = new PerformanceObserver((list) => {
  for (const entry of list.getEntries()) {
    console.log(` ${entry.name}, ${entry.startTime}`);
  }
});

observer.observe({ type: 'resource' });
```

Типы записей PerformanceObserver

Тип (entryType)	Описание
resource	Загрузка ресурсов: <code>img</code> , <code>fetch</code> , <code>script</code> , и др.
navigation	Полные данные о загрузке страницы (TTFB, redirect, DOM и др.)
mark	Пользовательские метки времени (<code>performance.mark()</code>)
measure	Промежутки между метками (<code>performance.measure()</code>)
paint	Ранние метрики отрисовки: <code>first-paint</code> , <code>first-contentful-paint</code>
first-input	Время отклика на первое взаимодействие (FID)
event	Задержки обработки событий (например, <code>click</code> , <code>input</code>)

Вычисление скорости ресурса

```
const entries = list.getEntries();

const lastEntry = entries[entries.length - 1];

if (lastEntry.initiatorType === 'fetch') {

    const size = lastEntry.transferSize;

    const duration = lastEntry.duration;

    const speed = size / duration;

    const isConnectionFast = speed > SLOW_CONNECTION_SPEED;

}
```

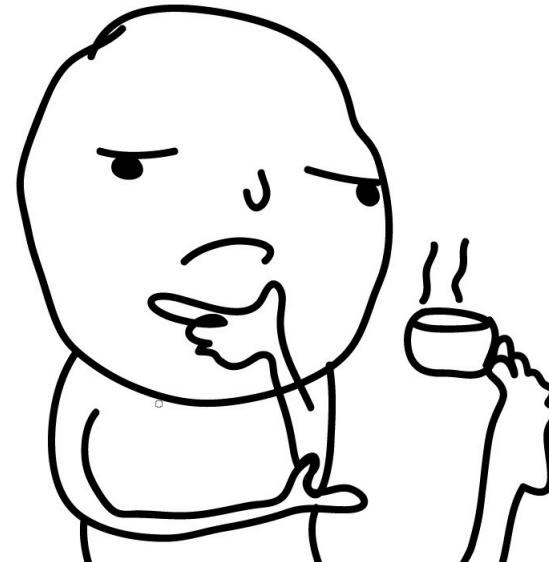
Фильтрация записей

- Не учитываем записи с duration < 100ms
- Не учитываем записи объёмом меньше 1536 байт

Что же считать «низкой» скоростью?

Зависит от:

- типа вашего приложения
- контента
- сценариев использования



Оценка скорости соединения

Скорость	Примерная оценка
< 50 КБ/сек	Очень медленно (пользователь заметит лаги)
50–150 КБ/сек	Замедленное поведение, страдает UX
150–500 КБ/сек	Приемлемо для большинства действий
> 500 КБ/сек	Хорошее подключение

WebSockets

- **WebSocket не попадает в список ресурсов**, отслеживаемых через Resource Timing API
- Это постоянное соединение, а Resource Timing предназначен для одноразовых загрузок ресурсов

WebSockets

Метрики WebSockets можно отслеживать вручную:

```
const wsStart = performance.now();

const socket = new WebSocket('wss://...');

socket.addEventListener('open', () => {

    const wsConnected = performance.now();

    console.log(`Connected in ${wsConnected - wsStart}ms`);

});
```

Что может пойти не так?

Проблемы с CORS

По умолчанию, если ресурс загружается с другого домена, то

`PerformanceResourceTiming` не раскроет многие поля:

```
console.log(entry.transferSize); // будет 0
```

```
console.log(entry.responseEnd); // тоже 0
```

Проблемы с CORS

Решение 1: ✓ Разрешить явно доступ к метрикам
(если у вас есть доступ)

Timing-Allow-Origin: <https://yourdomain.com>

или

Timing-Allow-Origin: *

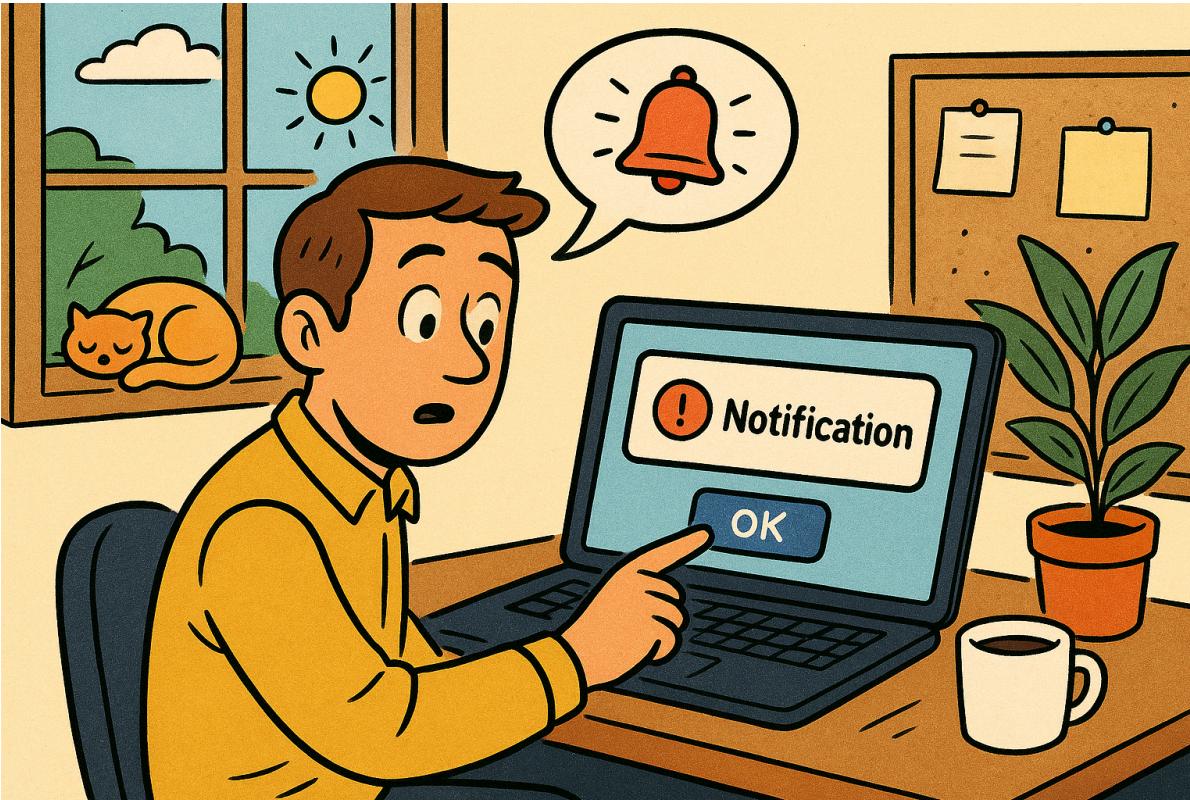
Проблемы с CORS

Решение 2:  Не учитываем запросы на другой домен

```
if (entry.name.startsWith(window.location.origin)) {  
    // считаем скорость  
}
```

Как наш сервис должен реагировать?

Показываем уведомление



Отображение уведомлений



Сеть нестабильна: возможны скачки и падения

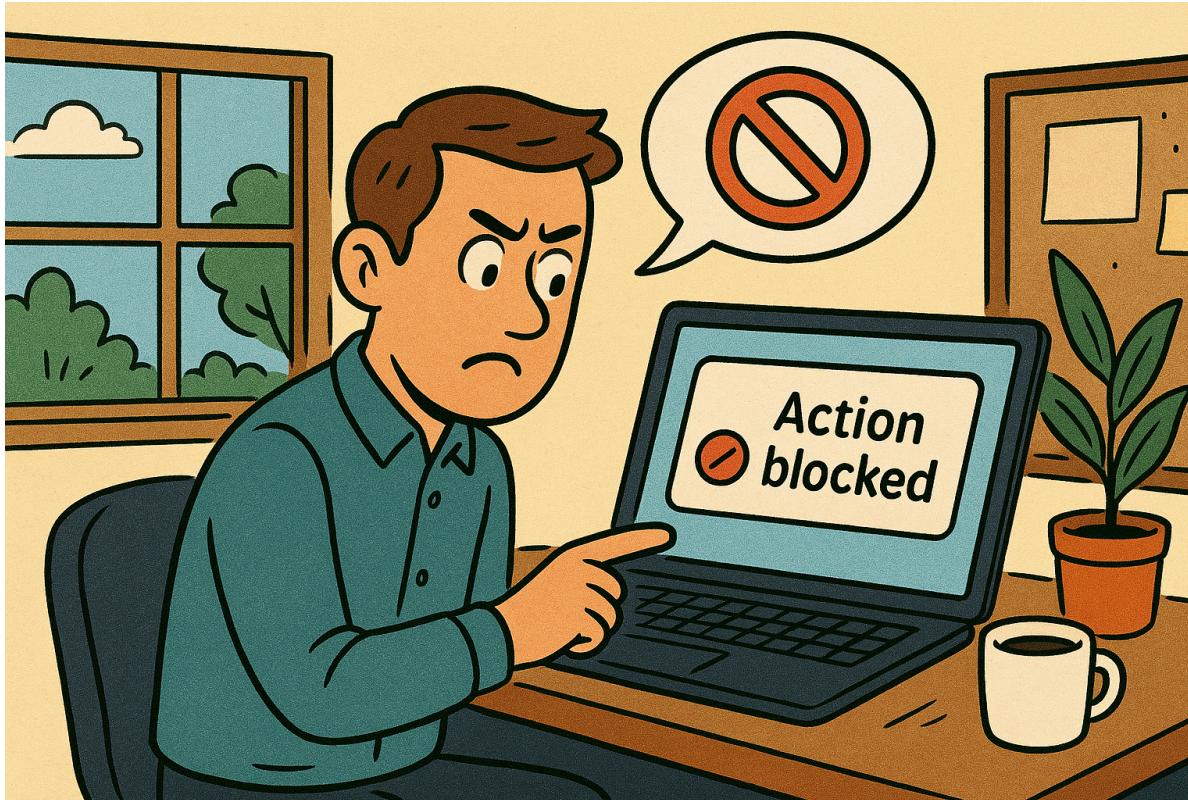


Замеры неточные: 1 медленный ресурс \neq медленная сеть



Уведомляем только по делу: когда это влияет на UX

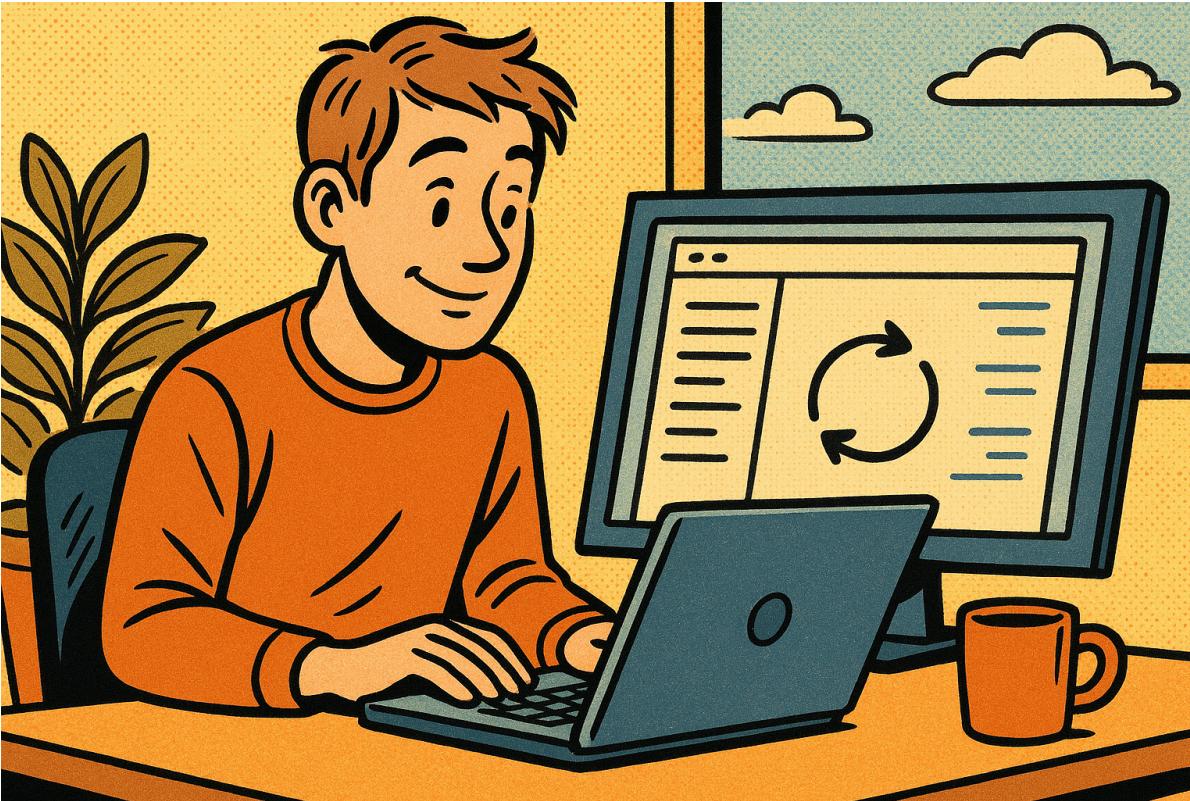
Блокируем интерактивные действия, которые могут привести к ошибке



Подгружаем облегчённую версию контента

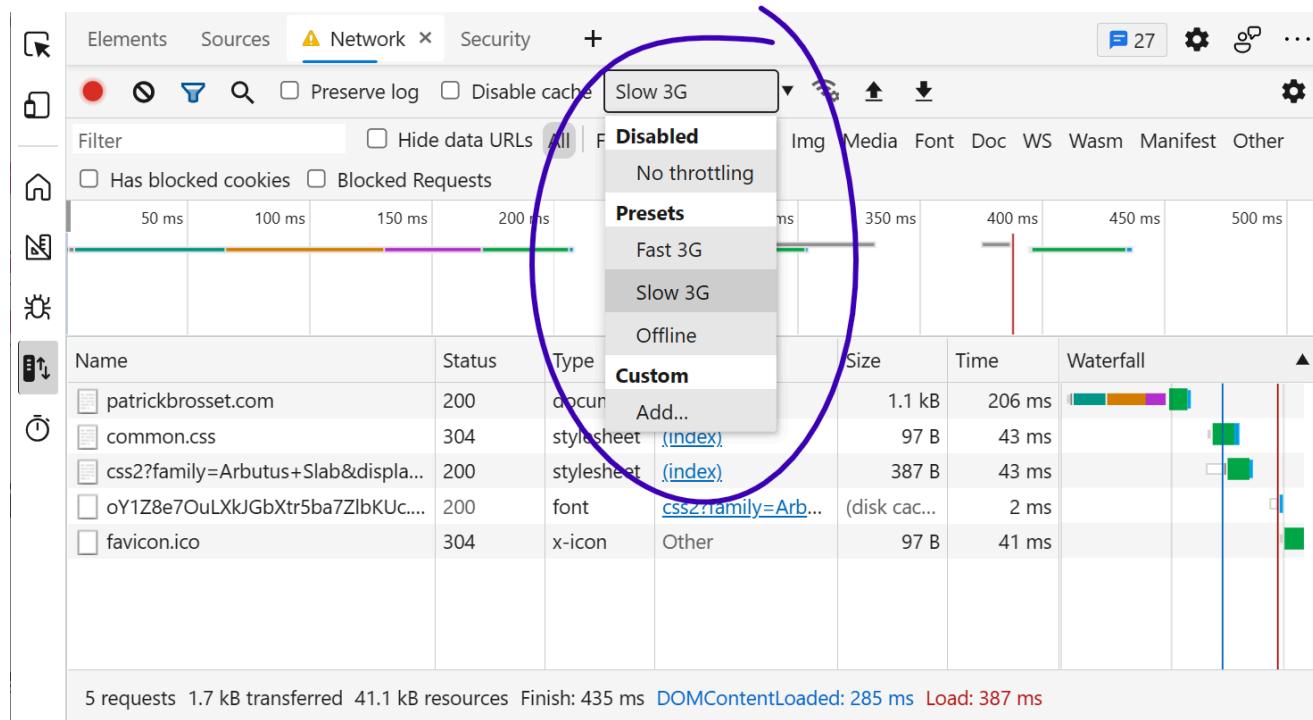


Начинаем повторную отправку запросов



Как же это всё проверить?

Эмуляция в браузере



The screenshot shows the Network tab in the Chrome DevTools. A purple circle highlights the dropdown menu for network throttling, which is currently set to "Slow 3G". The menu also includes options like "Disabled", "No throttling", "Presets" (Fast 3G, Slow 3G, Offline), and "Custom". Below the menu, a waterfall chart displays the loading times for various resources. The table below the chart lists the requests:

Name	Status	Type	Size	Time	Waterfall
patrickbrosset.com	200	document	1.1 kB	206 ms	
common.css	304	stylesheet	97 B	43 ms	
css2?family=Arbutus+Slab&display=... oY1Z8e7OuLXkJGbXtr5ba7ZlbKUc....	200	stylesheet	387 B	43 ms	
favicon.ico	304	x-icon	(disk cac...)	2 ms	
			97 B	41 ms	

At the bottom of the Network tab, the stats are: 5 requests, 1.7 kB transferred, 41.1 kB resources, Finish: 435 ms, DOMContentLoaded: 285 ms, Load: 387 ms.

Выводы

- 📢 Пользователи должны знать о проблемах с сетью
- 🧭 **navigator.onLine** работает только когда точно нет сети
- 🚫 Network Information API не работает
- ⚠️ Используйте Resource Timing API
- ⚠️ Не забывайте про CORS

**Пожалуйста, оставьте
свой отзыв**

Павел Дадыкин
ROGII



<https://meloman4eg.github.io/network-issues-fc-2025/>

[@meloman4eg](https://twitter.com/meloman4eg)



**Frontend
Conf**
2025