



FRONTEND 2019



ЦРИТО
Центр Развития
ИТ-Образования



МФТИ

@ mail.ru
group



Технологии



- React 16.8 [[lifecycle methods](#), [context API](#), [suspense](#), [React hooks](#)]
- [JAMstack](#)
- [squash](#) & [quicklink](#) by Google
- Фрэймворки: [RE:DOM](#), [Radi.js](#), [DisplayJS](#), [Stimulus](#), [Omi](#), [Quasar](#)
- Альтернатива Electron – [Neutralinojs](#)
- [Babel 7](#)
- [TypeScript](#) набирает популярность
- Встроенные карусели [scroll-snap](#)



[<https://jamstack.org/>]

JAMSTACK

JAMstack – методология создания статических сайтов.

JavaScript – интерактивный сайт

APIs – источник информации

Markups – шаблонизатор, создающий разметку страницы

Static Site Generators



Static Site Generators [SSG] - утилиты для создания статических сайтов

[HUGO](#)

[GATSBY](#)

[NEXT.JS](#)

Пример: <https://github.com/jamstack/jamstack.org> [HUGO]



- **Payment Request API** — cross-browser standard for Payment transaction process
- **Web Bluetooth API** — communications over GATT with nearby user-selected Bluetooth devices
- **WebAssembly** — new safe, portable, low-level code format for browsers
- **MediaRecorder API** — MediaStream Recording
- **WebRTC** — Real-time Peer-to-peer Communications Between Browser

Payment Request API



CANIUSE

Payment Request API - CR

Usage % of all users ?
Global 5.48% + 78.67% = 84.15%

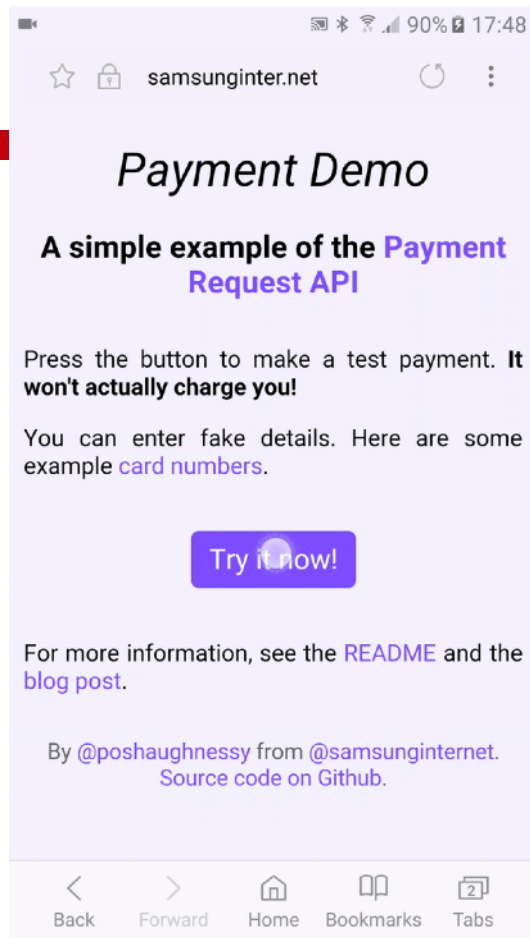
Payment Request is a new API for the open web that makes checkout flows easier, faster and consistent on shopping sites.

Current aligned Usage relative Date relative Apply filters Show all ?

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
	7 17		7 72		7 11.4				4
			7 73	7 12	7 12.1				7 8.2
11	7 18	8 66	7 74	12.1	12.2	all	5 74	11.8	7 9.2
	75	8 67	75	TP					
		8 68	76						
			77						

Payment Request API

Payment Request API —
позволяет организовать оплату
на сайте, используя нативный UI
устройства, с указанием деталей
карты, вариантами доставки,
адресами и контактными
данными — developers.google.com



Web Bluetooth API



CANIUSE

Web Bluetooth 📄 - UNOFF

Usage % of all users ?
Global 67.13%

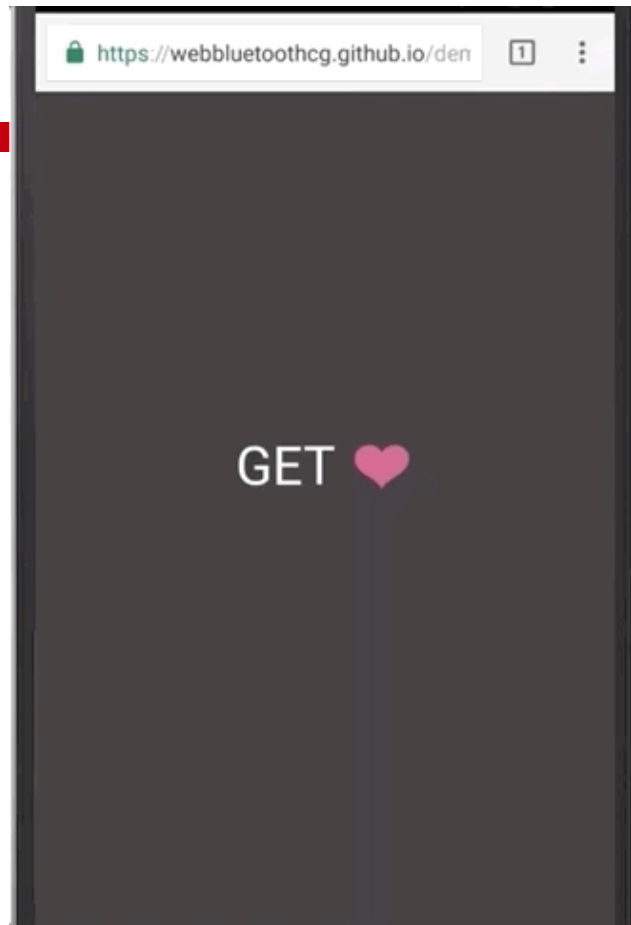
Allows web sites to communicate over GATT with nearby user-selected Bluetooth devices in a secure and privacy-preserving way.

Current aligned Usage relative Date relative Apply filters Show all ?

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			1 49						
			4 72		11.4				4
	17		4 73	12	12.1				8.2
11	18	66	4 74	12.1	12.2	all	4 74	11.8	9.2
	4 75	67	4 75	TP					
		68	4 76						
			4 77						

Web Bluetooth API

Web Bluetooth API —
предоставляет методы
взаимодействия браузера с
удалёнными Bluetooth LE
устройствами
— [developers.google.com](https://developers.google.com/web/updates/2018/07/web-bluetooth)



WebAssembly



CANIUSE

WebAssembly 📄 - OTHER

Usage % of all users ⌵ ?
Global 85.8%

WebAssembly or "wasm" is a new portable, size- and load-time-efficient format suitable for compilation to the web.

Current aligned Usage relative Date relative Apply filters Show all ?

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			72		11.4				4
	17		73	12	12.1				8.2
11	18	66	74	12.1	12.2	all	74	11.8	9.2
	75	67	75	TP					
		68	76						
			77						



WebAssembly (wasm) — эффективный низкоуровневый байт-код, предназначенный для исполнения в браузере.

WebAssembly представляет собой переносимое абстрактное синтаксическое дерево, обеспечивающее как более быстрый парсинг, так и более быстрое выполнение кода, чем JavaScript — developers.google.com

редактирование изображений/видео/музыки, математические вычисления, игры



- Бинарный формат
- НЕ язык программирования, а байт-код
- **Загружается в браузер и выполняется в браузере** — формально, WebAssembly выполняется JavaScript-движком, а не самим браузером, поэтому есть и другие варианты исполнения, например, под Node.js
- Исполняется виртуальной машиной
- НЕ имеет ничего общего с WEB, кроме того что общается с внешним миром через JavaScript

MediaRecorder API



CANIUSE

MediaRecorder API - WD

Usage % of all users ?
Global 73.36%

The MediaRecorder API (MediaStream Recording) aims to provide a really simple mechanism by which developers can record media streams from the user's input devices and instantly use them in web apps, rather than having to perform manual encoding operations on raw PCM data, etc.

Current aligned Usage relative Date relative **Apply filters** Show all ?

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			72		11.4				4
	17		73	12	12.1				8.2
11	18	66	74	12.1	12.2	all	74	11.8	9.2
	75	67	75	TP					
		68	76						
			77						



MediaRecorder API — позволяет получать доступ к камере и микрофону устройства через JavaScript API
— developers.google.com

MediaRecorder API



```
navigator.mediaDevices.getUserMedia(constraints)
  .then(function (stream) {
    const recordedBlobs = [];
    const mediaRecorder = new MediaRecorder(stream, {
      mimeType: 'video/webm',
    });
    mediaRecorder.ondataavailable = function (event) {
      recordedBlobs.push(event.data);
    };
    mediaRecorder.start(1000);           // для сохранения видео
    videoElement.srcObject = stream;     // для вывода
    // ...
    mediaRecorder.stop();
  });
```


WebRTC



CANIUSE

WebRTC Peer-to-peer connections 📄 - WD

Method of allowing two users to communicate directly, browser to browser using the `RTCPeerConnection` API.

Usage % of all users ?
Global 86.82% + 2.15% = 88.97%
unprefixed: 81.61% + 2.15% = 83.76%

Current aligned	Usage relative	Date relative	Apply filters	Show all	?				
IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			72		11.4				4
	17		73	12	12.1				8.2
11	18	66	74	12.1	12.2	all	74	11.8	9.2
	75	67	75	TP					
		68	76						
			77						



WebRTC (RTCPeerConnection API) — API, предоставляемое браузером и позволяющее организовать P2P соединение и передачу данных напрямую между браузерами
— codelabs.developers.google.com

медиа-стриминг, видео-общение, онлайн-игры, чаты, распределённые файловые системы, снижение нагрузки на сервера при передаче данных



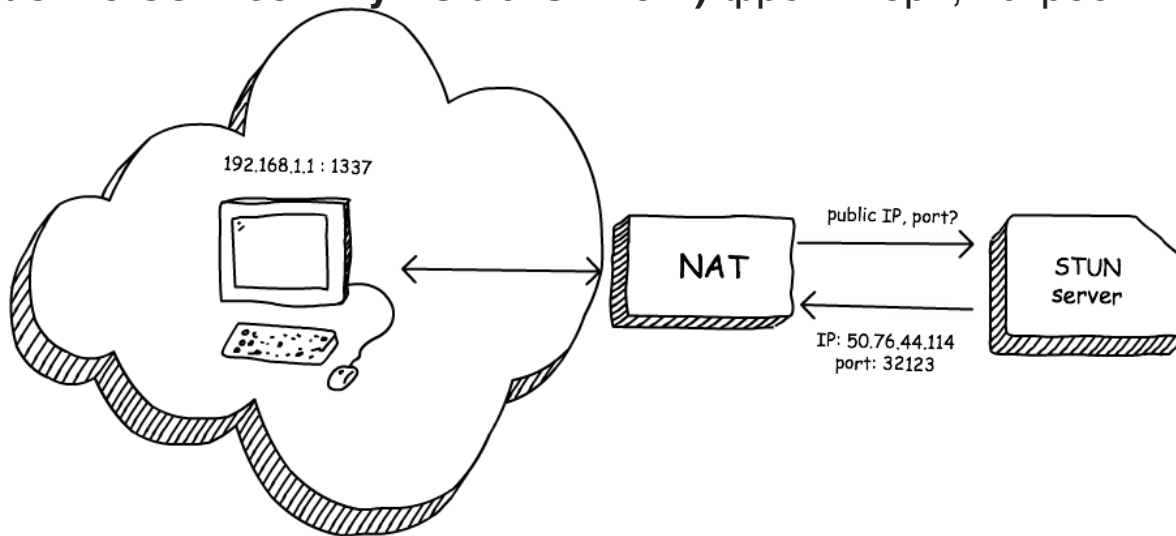
Для установления *peer-to-peer* оба клиента должны знать адрес и порт, по которому его оппонент слушает и может получить входящие данные. Для передачи этой информации необходим вспомогательный сервер — **signalling-сервер**



Кроме адресов и портов, клиенты должны договориться о **параметрах устанавливаемой сессии**. Например об использовании тех или иных кодеков и их параметров в случае аудио- и видеосвязи. Формат данных, описывающих всевозможные свойства соединения, называется **SDP — Session Description Protocol**



Для определения адреса клиента для WebRTC соединений используются дополнительные сервера **STUN** — **Session Traversal Utilities for NAT**. Задачу взаимодействия со STUN и задачу проверки доступности берет на себя **ICE** (**Interactive Connectivity Establishment**) фреймворк, встроенный в браузер





```
const iceServers = [{ url: 'stun:stun.l.google.com:19302' }];
const peerConnection = new RTCPeerConnection({iceServers}, constraints);

peerConnection.setConfiguration(configuration);
peerConnection.onicecandidate = function (event) {
    // функция, с помощью которой RTCPeerConnection отправляет на сервер
    // сообщения, которые сервер должен вернуть другому браузеру
};
peerConnection.onaddstream = function (event) {
    // вызывается, когда удалённая сторона присылает новый stream с данными
};
const rtpSender = RTCPeerConnection.addTrack(track, stream...);
```



JS специалист



- Веб протоколы: HTTP, HTTP/2, HTTPS, WebSocket
- JavaScript — es6+, es5 и TypeScript
- Возможности браузеров (HTML5, CSS)
- Кроссбраузерность (API, верстка)
- Устройство браузера изнутри
- Безопасность web-приложений (XSS, CSRF, cookies, CORS)
- Инфраструктура системы, деплой приложения
- Представление об UX
- Понимание server-side
- Общая картина разработки ПО



- Инструменты разработки: Node.js
- Инструменты сборки: Webpack, grunt, gulp, parcel
- Инструменты работы с CSS: PostCSS
- Модульные системы: ES6-модули, AMD, UMD
- DevTools браузеров

Нестандартные подходы



- Реактивное программирование [[RxJS](#)]
- Функциональное программирование [[Ramda](#)]



- [front end handbook](#)
- [state of js](#)
- [overreacted](#)
- [dev tips](#)
- [css weekly](#)
- [ecmascript daily](#)



- [webstandards \[tg\]](#)
- [Jake Archibald \[twitter\]](#)
- [Addy Osmani \[twitter\]](#)
- [Dan Abramov \[twitter\]](#)
- [Surma \[twitter\]](#)



- [Chrome Developers](#)
- [Frontend channel](#)
- [Технострим Mail.Ru Group](#)
- [Academind](#)
- [Frontend Masters](#)



- [moscowJS](#)
- [holyJS](#)



- [es 6 katas](#)
- [checkio](#)
- [leetcode](#)
- [topcoder arena](#)



Онлайн ресурсы

- [front-end-interview-handbook](#)
- [js garden](#)
- [awesome interview questions #javascript](#)
- [javascript interview questions](#)
- [30 seconds of code](#)

Книги

- [Cracking The Coding Interview 6](#)
- [Are You Smart Enough To Work In Google](#)



Спасибо за внимание!