

SERVICEWORKERS

E O FUTURO DAS APLICAÇÕES NO SEU BROWSER



EDUARDO MATOS

@eduardojmatos



LABORATÓRIO DE PROGRAMAÇÃO

*estamos contratando!

An aerial photograph of the city of Porto Alegre, Brazil, featuring a dense cluster of modern skyscrapers along a long, narrow peninsula that extends into a large body of water. The city is surrounded by green hills and a clear blue sky.

POA É A
SAN FRANCISCO DO
BRASIL!

O QUE É UM WORKER?



Primeira definição de Worker surgiu com o WebWorkers

<http://www.html5rocks.com/en/tutorials/workers/basics/>

BROWSER EXECUTANDO TAREFAS SEM BLOQUEAR A INTERFACE

Resumindo: scripts rodando em paralelo à pagina

EX: tarefas que exigem alto processamento

```
// main script
var worker = new Worker('worker.js');

worker.addEventListener('message', function(e) {
  console.log('Worker diz: ', e.data);
}, false);

worker.postMessage('Oi, eu sou o Worker!');

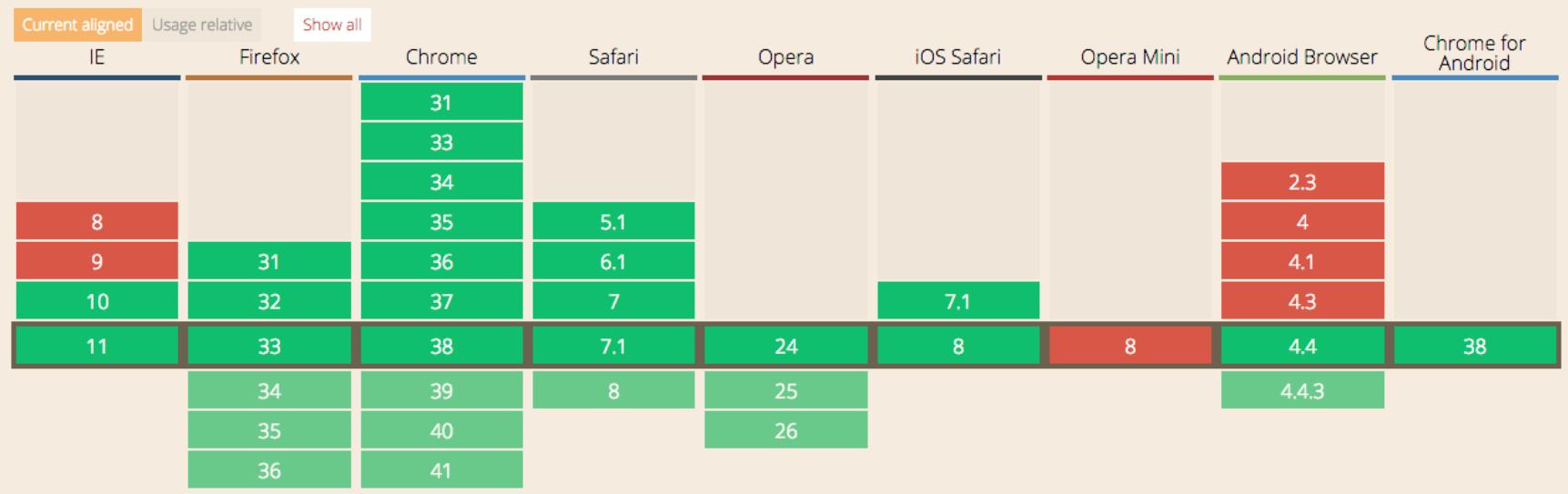
// worker.js
self.addEventListener('message', function(e) {
  self.postMessage(e.data);
}, false);
```

Web Workers - CR

Global

81.32%

Method of running scripts in the background, isolated from the web page



REGRAS COMUNS DE UM WORKER

- Não acessa o DOM
- Não acessa o window
- Não acessa o document
- Acessa o navigator
- Acessa o location (ready-only)

WHO ARE SERVICEWORKERS?

- Usa o mesmo conceito do WebWorker de isolar execução de um script
- Controla programaticamente o cache da sua aplicação
- Serve como um proxy da sua aplicação
- [futuramente] BackgroundSync (!!)
- [futuramente] Geofencing (!!!)
- [futuramente] Push Notifications (!!!!)
- [futuramente] Bluetooth (!!!!)



https://slightlyoff.github.io/ServiceWorker/spec/service_worker/index.html

W3C Editor's Draft



Service Workers

W3C Editor's Draft 15 October 2014

This version

https://slightlyoff.github.io/ServiceWorker/spec/service_worker/

Latest editor's draft

https://slightlyoff.github.io/ServiceWorker/spec/service_worker/

Previous version

<http://www.w3.org/TR/service-workers/>

Revision history

<https://github.com/slightlyoff/ServiceWorker/commits/master>

Participate

Discuss on public-webapps@w3.org (Web Applications Working Group)

[File bugs](#)

Editors

Alex Russell, Google, <slightlyoff@chromium.org>

Jungkee Song, Samsung Electronics, <jungkee.song@samsung.com>

Copyright © 2014 W3C® ([MIT](#), [ERCIM](#), [Keio](#), [Beihang](#)). All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.

OFFLINE CONTROL

Sorry NO
INTERNET Today



**VAI APPLICATION
CACHE!**

A forma de cachear arquivos parecia simples...

<http://diveintohtml5.info/offline.html>





APPCACHE IS A DOUCHEBAG!

<http://alistapart.com/article/application-cache-is-a-douchebag>

<http://sergiolopes.org/palestra-appcache-html5-offline>

<http://eduardomatos.me/appcache-manifest-e-serviceworker-as-partes-boas-e-ruins>

LÁ VEM O
SERVICEWORKERS



KEEP
CALM
CAUSE
LÁ VEM O
MARCOS



```
// main script
navigator.serviceWorker.register("/assets/worker.js").then(
  function ( serviceWorker ) {
    serviceWorker.postMessage("ServiceWorkers instalado com sucesso.");
  },
  function ( error ) {
    console.error("Ops.. não rolou a instalação do ServiceWorkers", error);
  });
});
```

Interceptando requests

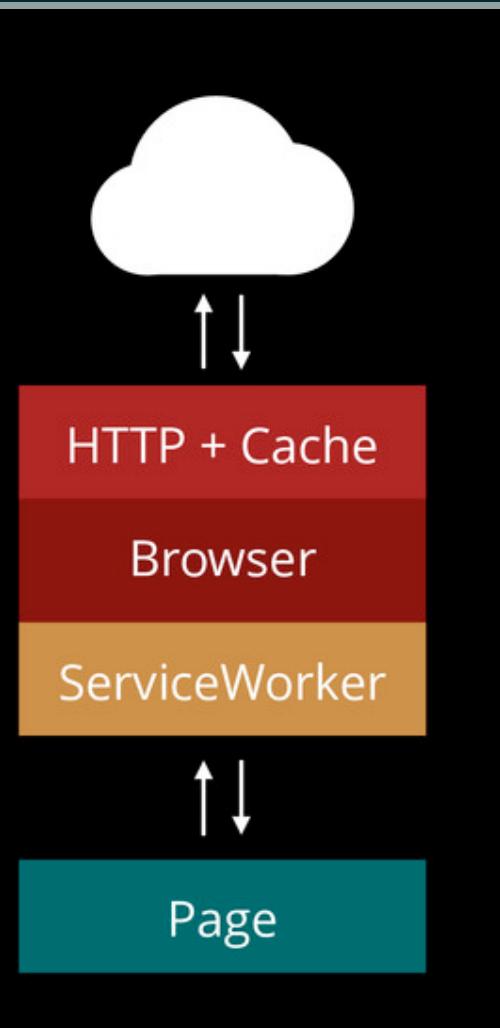
```
// assets/worker.js
var base = "https://meu.server.com";

var data_url = new URL("/data/posts.json", base) + "";

self.addEventListener("fetch", function(e) {
  var url = e.request.url;
  console.log(url);
  if (url == data_url) {
    e.respondWith(
      new Response(JSON.stringify({
        posts: { /* ... */ }
      }), {
        headers: { 'Content-Type': 'application/json' },
        status: 200
      })
    );
  }
});
```

```
https://meu.server.com/data/posts.json  
https://meu.server.com/assets/application.css  
https://meu.server.com/assets/application.js  
https://meu.server.com/assets/logo.jpg
```

ServiceWorker sits
between your page
and the browser's
network stack.



Cacheando seus assets

```
// assets/worker.js
self.version = 1;

var base = "https://meu.server.com"

// Listener para a instalação do ServiceWorkers
self.addEventListener("install", function ( e ) {
    var myResources = new Cache (
        base + "/index.html",
        base + "/assets/application.css",
        base + "/assets/application.js",
        base + "/assets/logo.png",
        base + "/data/posts.json",
        base + "/data/posts_fallback.json"
    );

    e.waitUntil(myResources.ready());
    caches.set("caches-" + self.version, myResources);
});
```

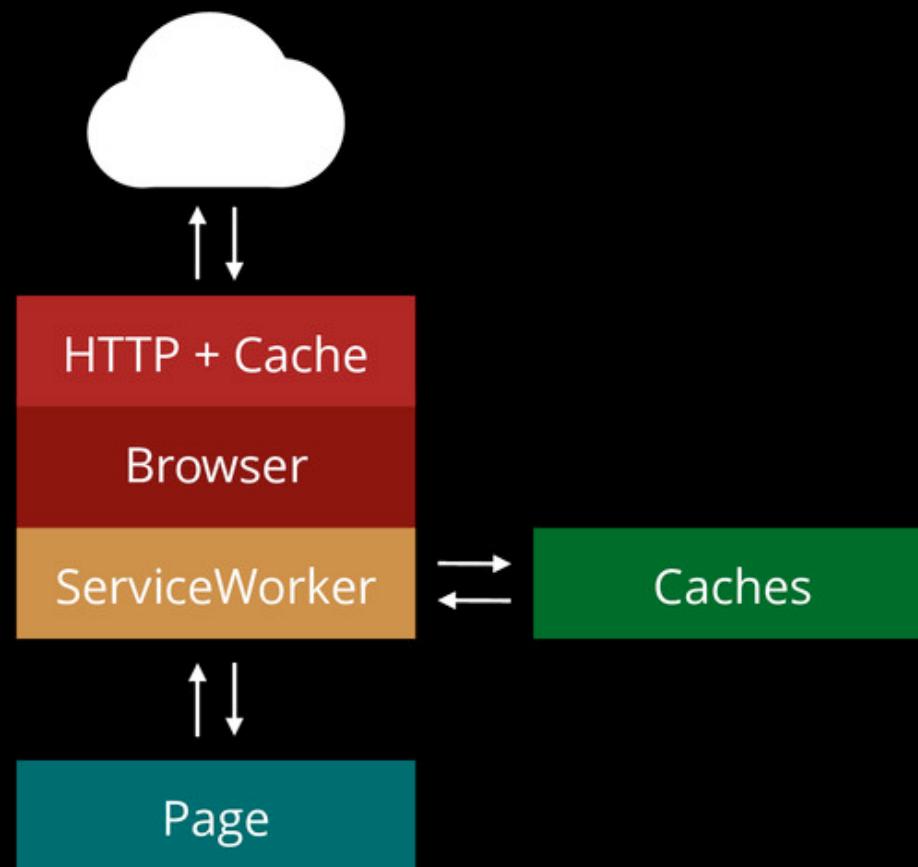
Request + Cache

```
// assets/worker.js
self.version = 2;

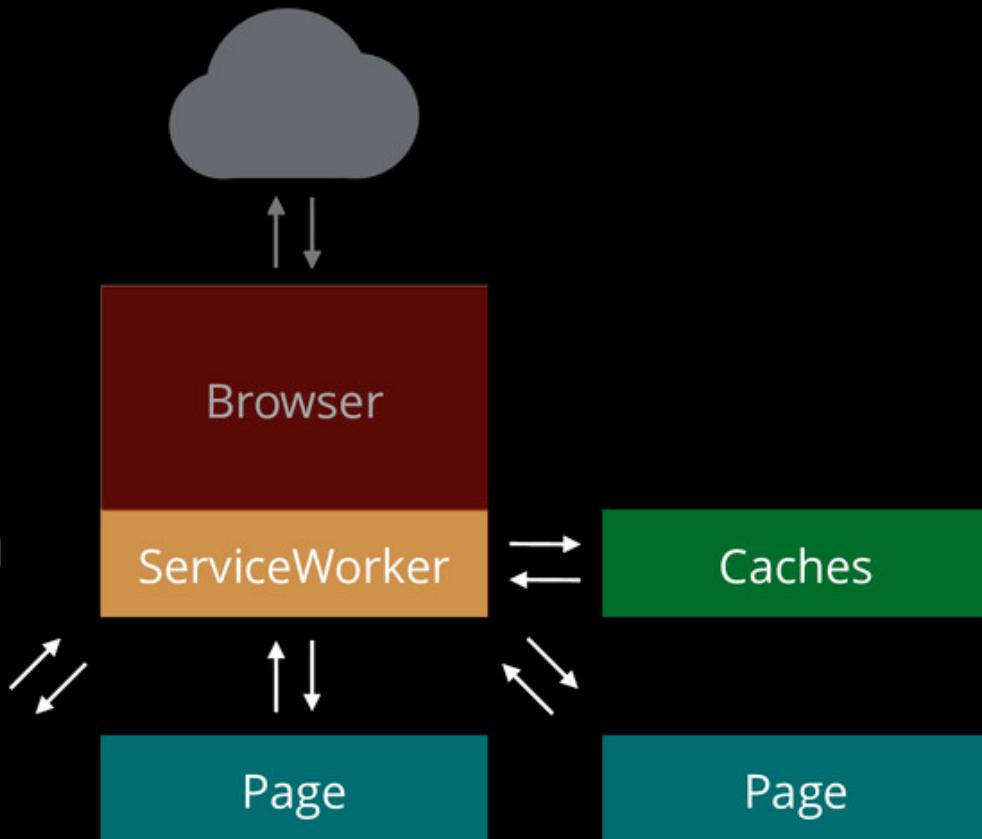
var base = "https://meu.server.com"
var data_url = new URL("/data/posts.json", base) + "";

self.addEventListener("fetch",
    function ( e ) {
        if (e.request.url == data_url) {
            e.respondWith(
                caches.match( e.request )
                    .catch( function () {
                        return e.default;
                    })
                    .catch( function () {
                        return caches.match("/data/posts_fallback.json");
                    })
            );
        }
    }
);
});
```

It all has
programmatic
access to a set of
durable caches.



1 to many, like an extension.



EXEMPLO DE SERVICEWORKERS

<https://jakearchibald.github.io/trained-to-thrill/>

E A SEGURANÇA DISSO?

- Funciona apenas com https
- Escopo por domains

BACKGROUND SYNC

Ainda muito recente

```
// main script
navigator.serviceWorker.register("/assets/worker.js");

navigator.serviceWorker.ready.then(function (sw) {
    // Returns a Promise
    navigator.sync.register(
        "my_data_sync",
        {
            minInterval: 86400 * 1000,           // ms, default: heuristic
            repeating: true,                  // default: false
            data: '',                         // default: empty string
            description: '',                 // default: empty string
            lang: '',                         // default: document lang
            dir: ''                          // default: document dir
        }
    ).then(function() { // Success
        // No resolved value
        // Success, sync is now registered
    },
    function() { // Failure
        // If no SW registration
        // User/UA denied permission
        // Sync id already registered
    });
});
```

```
// assets/worker.js
self.onsync = function(event) {
  var data = JSON.parse(event.data);

  if (event.id === "my_data_sync") {
    if (data.whatever === "foo") {
      // rejection is indication that the UA should try
      // later (especially when network is ok)
      event.waitUntil(doAsyncStuff());
    }
  } else {
    // Garbage collect unknown syncs (perhaps from older pages).
    navigator.sync.unregister(event.id);
  }
};
```

GEOFENCING

Nem a spec foi definida.

BLUETOOTH

Spec ainda engatinhando

<https://github.com/WebBluetoothCG/web-bluetooth>

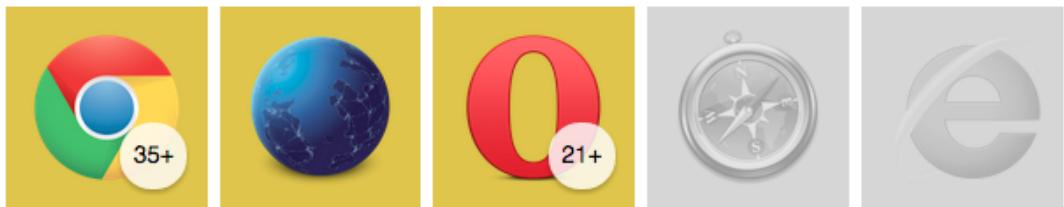
<http://www.w3.org/community/web-bluetooth>

ONDE FUNCIONA?

<https://jakearchibald.github.io/isserviceworkerready/>

navigator.serviceWorker

Namespace for page-side ServiceWorker API. [Spec.](#) [Test.](#)



Firefox: Requires a [nightly build](#) and `dom.serviceWorkers.enabled` in `about:config`.

Chrome & Opera: Requires `#enable-experimental-web-platform-features` in `about:flags`.

PORQUE ESSE HYPE TODO?

- JavaScript chegando no app nativo
- Evolução e maior controle das aplicações
- Acesso a funcionalidades do sistema do usuário (bluetooth, push notifications...)



OBRIGADO!

link da palestra: <http://bit.ly/serviceworkers-tdc2014>

meu twitter: [@eduardojmatos](https://twitter.com/eduardojmatos)

fontes: <http://bit.ly/serviceworkers-links>