



Por: Felipe N. Moura

Google Developer Expert





Google Developer Expert

Fundador da  
BrazilJS e da Nasc



Google Developer Expert

Fundador da  
BrazilJS e da Nasc

Cerca de 12 anos de  
desenvolvimento Web





Google Developer Expert

Fundador da  
BrazilJS e da Nasc

Cerca de 12 anos de  
desenvolvimento Web

Apaixonado por JS!!



# Service Workers

a seu serviço



# Service workers...Why?!

- Controle sobre requisições
- Controle sobre cache
- Controle sobre redirecionamentos



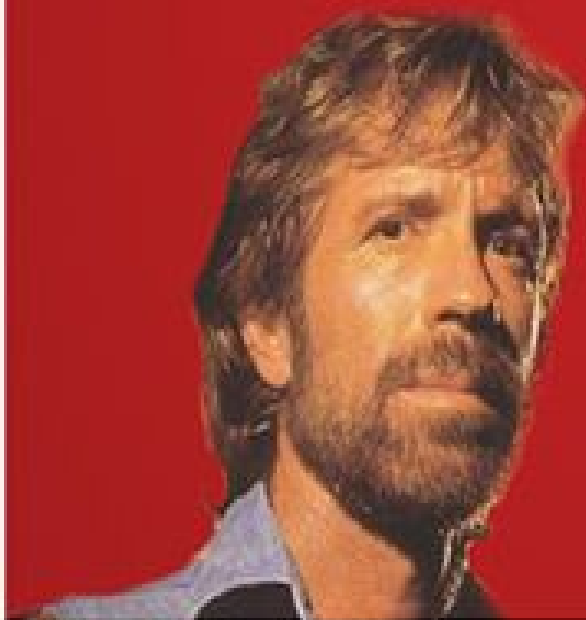
More POWER, babe!!!



# Special powers

- Gerenciar redireccionamientos

Looks like  
**CHUCK NORRIS**  
broke this page.



...It may never recover.

[Go Back](#)

[Back Home](#)

# Special powers

- Gerenciar redirecionamentos
- Interceptar requisições





# Special powers

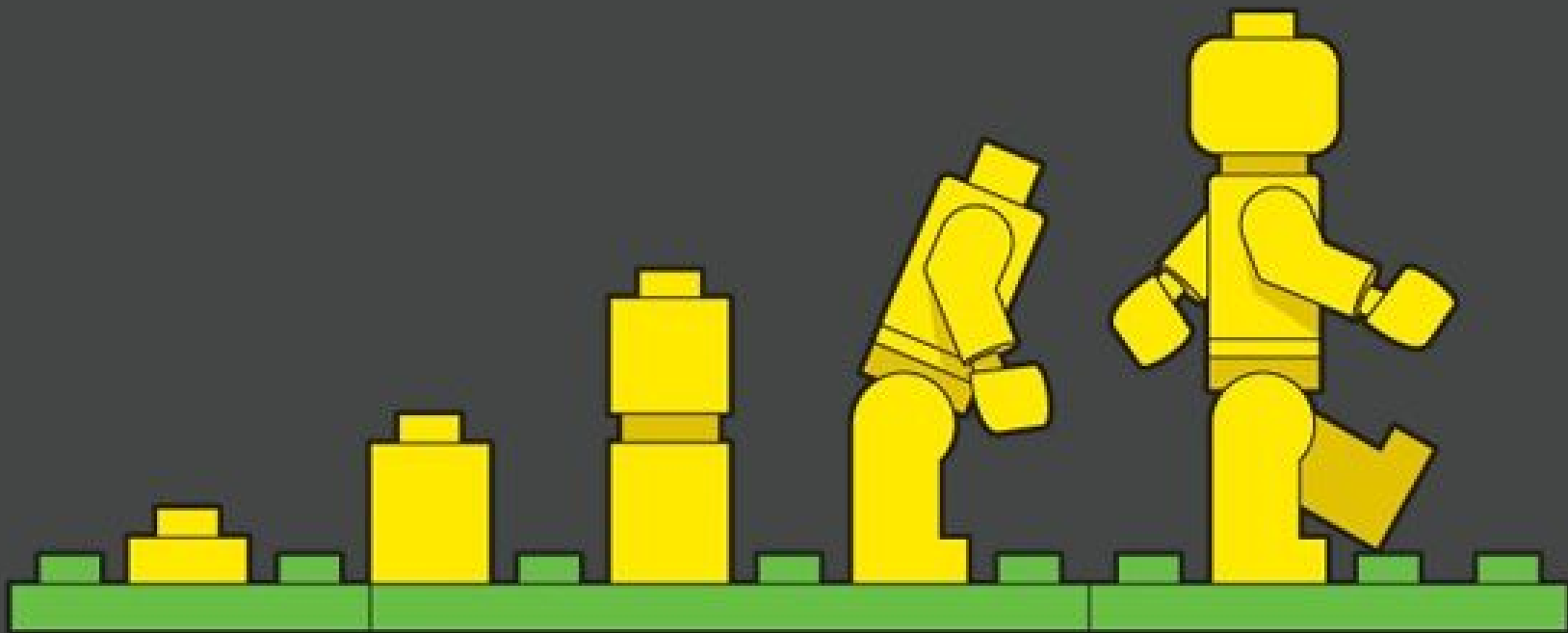
- Gerenciar redirecionamentos
- Interceptar requisições
- Controle de cache





# Special powers

- Gerenciar redirecionamentos
- Interceptar requisições
- Controle de cache
- Controle de versão



# Special powers

- Gerenciar redirecionamentos
- Interceptar requisições
- Controle de cache
- Controle de versão
- Evolução progressiva



PROGRESSIVA

# How?

- Baseado em tecnologias Web
- Tecnologias Open source
- Assíncrono (promises)
- Seguro (https, e no escopo do service worker)

# Evolução do próprio Service Worker

- Promises
- Cache API
- Fetch
- Web Notifications

WEB

# What for?

- Suporte em diferentes plataformas



# What for?

- Suporte em diferentes plataformas
- Funciona mesmo offline, ou quando a internet está intermitente

# What for?

- Suporte em diferentes plataformas
- Funciona mesmo offline, ou quando a internet está intermitente
- Torna-se "Instalável" no dispositivo do usuário

# What for?

- Suporte em diferentes plataformas
- Funciona mesmo offline, ou quando a internet está intermitente
- Torna-se "Instalável" no dispositivo do usuário
- Muito mais performático

APP

# PROGRESSIVE WEB APP



Internet

No

Brasil

06/04/2016 10h02 - Atualizado em 06/04/2016 14h14

# Smartphone passa PC e vira aparelho nº 1 para acessar internet no Brasil

Celular está em 80% das casas e PC, em 76,6%; dado do IBGE é de 2014. Pnad mostra que, pela 1ª vez, internet chega a mais de 50% das casas.

**Helton Simões Gomes**  
Do G1, em São Paulo



FACEBOOK



# Velocidade de internet no Brasil está 28% abaixo da média

Reprodução/sxc.hu

Internet: Brasil tem velocidade média pior do que a da Argentina

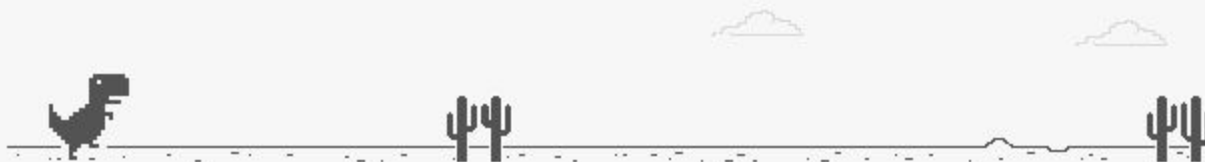


Lucas Agrela, de  
EXAME.com

Siga-me

A velocidade média da internet global é de 5 Mbps, mas o Brasil tem somente 3,6 Mbps.

De acordo com dados do relatório trimestral da Akamai, chamado State of the Internet, ao qual EXAME.com teve acesso, o país está na 90ª posição no ranking mundial, atrás de países como Uruguai (5,9 Mbps) e Argentina (4,7 Mbps).



## There is no Internet connection

You can try to diagnose the problem by taking the following steps:

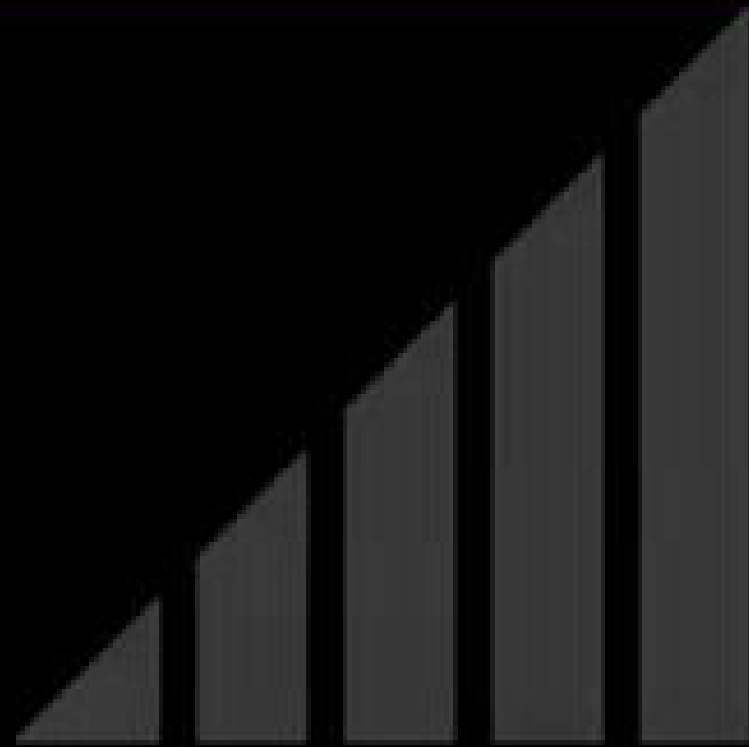
Go to **Applications > System Preferences > Network > Assist me** to test your connection.

Try:

- Checking the network cable or router
- Resetting the modem or router
- Reconnecting to Wi-Fi

ERR\_INTERNET\_DISCONNECTED

DETAILS



**Lie-Fi**







- 🏠 Home
- 👤 My Channel
- 🔥 Trending
- 📺 Subscriptions 24
- 🕒 History
- 🕒 Watch Later

LIBRARY

- ≡ working
- ≡ FWorld
- ≡ felipenmoura.com

Show more

SUBSCRIPTIONS

Filters ▾

About 4,360,000 results



5:24

### Top 5 Best Ways To Destroy Your iPhone 6!

TechRax ✓

1 year ago • 4,590,271 views

The ultimate life hacks all compiled to successfully ruin/destroy your iPhone 6. Works on any smartphone too! FACEBOOK: ...



2:02

### How to destroy your phone, featuring Sen. Lindsey Graham

CBSN

11 months ago • 6,858 views

After Donald Trump gave out Lindsey Graham's phone number at a campaign event, the South Carolina senator made a video ...

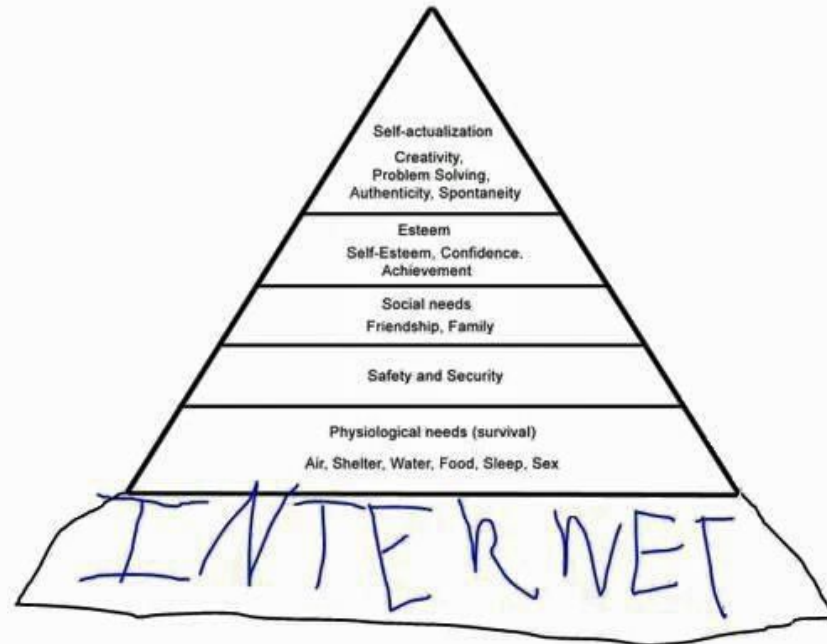


### What Happens If You Shoot an iPhone 6?

TechRax ✓

1 year ago • 20,133,965 views

# Necesidades básicas



O celular, é apenas  
mais uma vítima



# O culpado é o desenvolvedor que...

- Não minifica scripts e estilos
- Não otimiza imagens
- Tem mais banners que conteúdo
- Tem o DOM bagunçado
- Páginas não responsivas



# Mas além disto...

"Uma PWA é mais do que apenas uma página com boa performance."

# Mas além disto...

"Uma PWA é mais do que apenas uma página com boa performance."

-- by Felipe N. Moura

Let's CODE!

But first...

The theory



# Teoria

"A teoria é a diferença entre saber, e SABER"

# Teoria

"A teoria é a diferença entre saber, e SABER"  
-- by Felipe N. Moura

# Service workers

- Rodam em escopo próprio
- Uma vez registrados, rodam em "background"
- Exigem HTTPS (ou localhost)
- Somente podem interceptar requests no próprio escopo





# Worker lifecycle

## INSTALLING

This stage marks the beginning of registration. It's intended to allow to setup worker-specific resources such as offline caches.

 `install`



-  Use **`event.waitUntil()`** passing a promise to extend the installing stage until the promise is resolved.
-  Use **`self.skipWaiting()`** anytime before activation to skip installed stage and directly jump to activating stage without waiting for currently controlled clients to close.

# Service Worker lifecycle

## INSTALLED

The service worker has finished its setup and it's waiting for clients using other service workers to be closed.





## ACTIVATING

There are no clients controlled by other workers. This stage is intended to allow the worker to finish the setup or clean other worker's related resources like removing old caches.

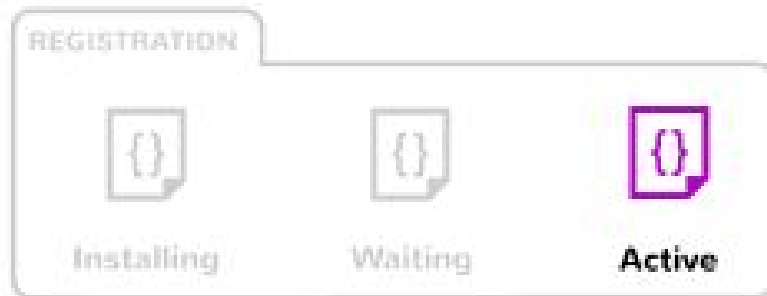


 **activate**

-  Use **event.waitUntil()** passing a promise to extend the activating stage until the promise is resolved.
-  Use **self.clients.claim()** in the activate handler to start controlling all open clients without reloading them.

## ACTIVATED

The service worker can now handle functional events.



## REDUNDANT

This service worker is being replaced by another one.





Let's CODE!

Let's CODE!

Now, for real!

# HTML (index.html)

```
<script>
  if ('serviceWorker' in navigator) {
    navigator.serviceWorker.register('/sw.js')
      .then(function(reg) {
        console.log('Alright, yeah!', reg.scope);
      }).catch(function(error) {
        console.log('Oops!', error);
      });
  }
</script>
```



HTML (index.html)

DONE



Browser que não suportam Service Worker,  
nem saberão o que estão perdendo!

# sw.js

```
self.addEventListener('install', function(event) {  
  // event is an ExtendableEvent  
  event.waitUntil(new Promise(resolve, reject){  
    // do something  
    resolve();  
  });  
});  
  
self.addEventListener('activate', function(event) {  
  event.waitUntil(new Promise(resolve, reject){  
    // do something  
    resolve();  
  });  
});
```



## Application

- Manifest
- Service Workers
- Clear storage

## Storage

- ▶  Local Storage
- ▶  Session Storage
- ▶  IndexedDB
- ▶  Web SQL
- ▶  Cookies

## Cache

- Cache Storage
- Application Cache

## Frames

-  top

## Service Workers

- ☐
- Offline
- ☐
- Update on reload
- ☐
- Bypass for network
- ☒
- Show all

<http://localhost:8888/>

[Update](#) [Push](#) [Sync](#) [Unregister](#)

Source [dsw.js?dsw-manager](#)

Received 7/17/2016, 10:58:53 PM

Status ● #2014 activated and is running [stop](#)

Clients <http://localhost:8888/> [focus](#)

# sw.js

```
self.addEventListener('fetch', event=>{  
  const url = new URL(event.request.url);  
  
  event.respondWith(fetch(event.request)  
    .then(treatSuccess)  
    .catch(treatFailure)  
  );  
});
```

# SW.js

```
function treatSuccess (response) { return response; }  
function treatFailure (response) { return response; }
```

\* Deve retornar um objeto *Response* para o método *respondWith*.

# Sw.js - redirects

```
self.addEventListener('fetch', event=>{
  const url = new URL(event.request.url);
  const NEWER = '/index.html';

  if (url.pathname.match(/older\/.*/)) {
    return event.respondWith(fetch(NEWER));
  }
  event.respondWith(fetch(event.request));
});
```

# Sw.js - redirects (404)

```
self.addEventListener('fetch', event=>{
  const url = new URL(event.request.url);
  const NOT_FOUND = '/my-404.html';

  event.respondWith(fetch(event.request)
    .then(response=>{
      if (response.status == 404) {
        return fetch(NOT_FOUND);
      }
      return response;
    })
  );
});
```





- Redireccionamiento via JavaScript
- 404 customizado, via JavaScript





Cool, or what?!



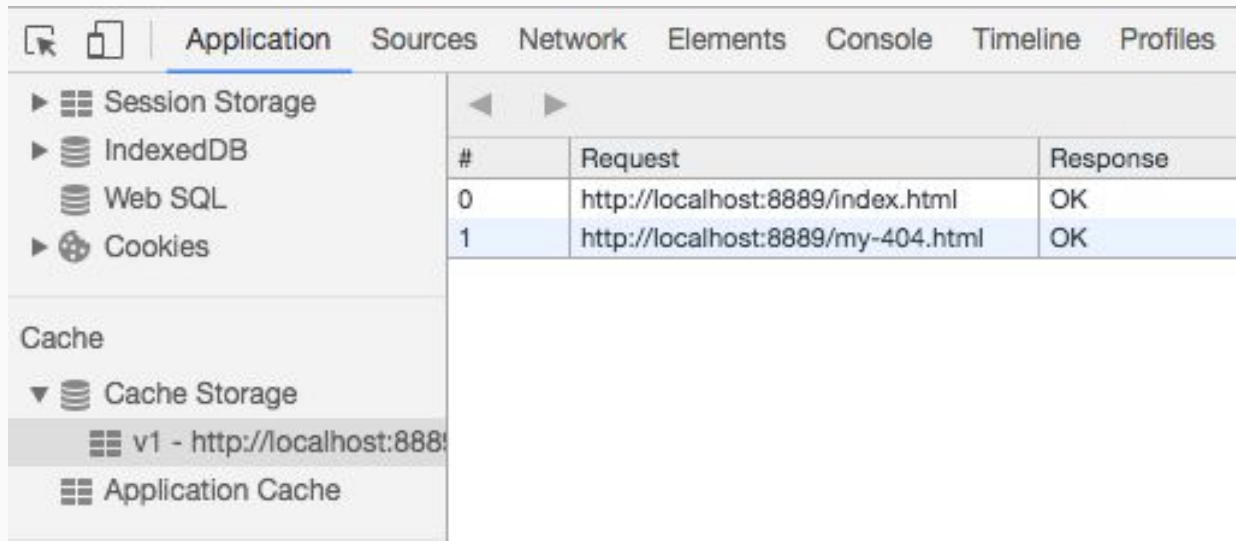
Hora de  
mergulhar fundo!

# Cache

```
const CACHE_GROUP = 'v1';

self.addEventListener('install', function(event) {
  event.waitUntil(
    caches.open(CACHE_GROUP)
      .then(cache=>{
        return cache.addAll([
          'index.html',
          'my-404.html'
        ]);
      })
  );
});
```

# Cache

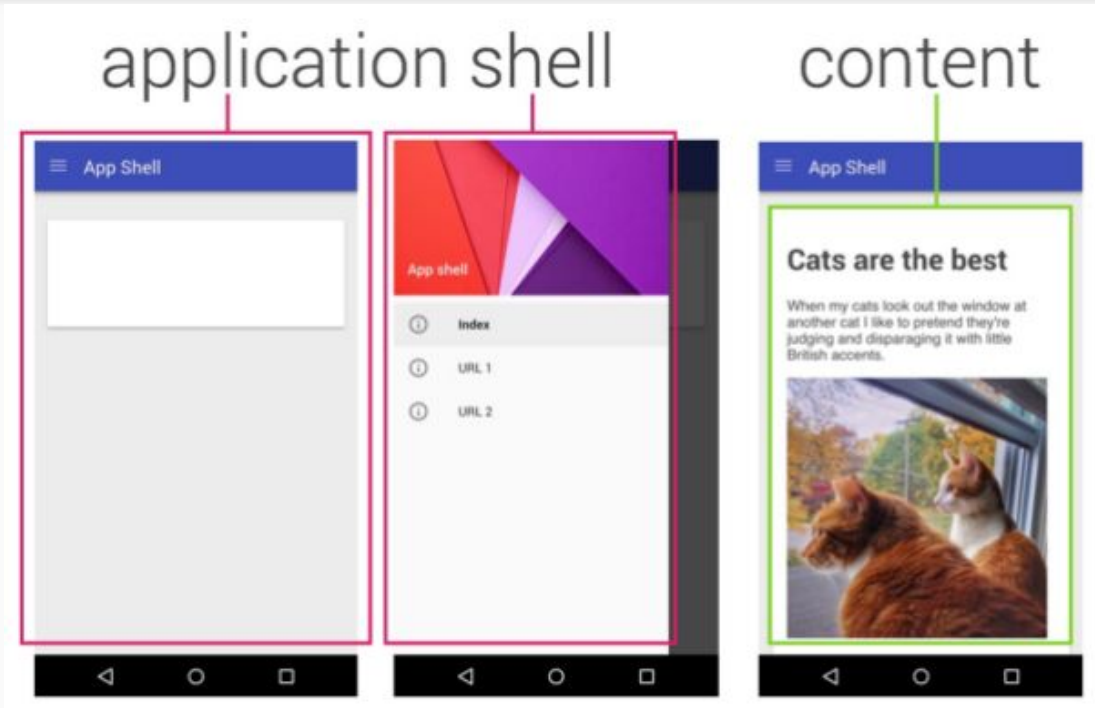


The screenshot shows the Chrome DevTools Application tab. The left sidebar contains a tree view of storage areas: Session Storage, IndexedDB, Web SQL, Cookies, Cache, and Application Cache. The 'Cache' section is expanded, showing 'Cache Storage' with a sub-entry 'v1 - http://localhost:8889/'. The main panel displays a table of network requests and responses.

#	Request	Response
0	http://localhost:8889/index.html	OK
1	http://localhost:8889/my-404.html	OK



# Application Shell



# Fetch from Cache

```
self.addEventListener('fetch', event=>{  
  const url = new URL(event.request.url);  
  const NOT_FOUND = '/my-404.html';  
  
  event.respondWith();  
  
});
```

# Fetch from Cache

```
self.addEventListener('fetch', event=>{  
  const url = new URL(event.request.url);  
  const NOT_FOUND = '/my-404.html';  
  
  event.respondWith(  
    caches.match(event.request)  
  );  
  
});
```



# Fetch from Cache

```
self.addEventListener('fetch', event=>{
  const url = new URL(event.request.url);
  const NOT_FOUND = '/my-404.html';

  event.respondWith(
    caches.match(event.request)
      .then(result=>{
        return result;
      })
  );
});
```

# Fetch from Cache

```
self.addEventListener('fetch', event=>{
  const url = new URL(event.request.url);
  const NOT_FOUND = '/my-404.html';

  event.respondWith(
    caches.match(event.request)
      .then(result=>{
        return result || fetch(event.request);
      })
  );
});
```

# Fetch from Cache

```
self.addEventListener('fetch', event=>{
  const url = new URL(event.request.url);
  const NOT_FOUND = '/my-404.html';

  event.respondWith(
    caches.match(event.request)
      .then(result=>{
        return result || fetch(event.request)
          .then(response=>{
            if (response.status == 200) {
            }
            return response;
          });
      });
  );
});
```

# Fetch from Cache

```
self.addEventListener('fetch', event=>{
  const url = new URL(event.request.url);
  const NOT_FOUND = '/my-404.html';

  event.respondWith(
    caches.match(event.request)
      .then(result=>{
        return result || fetch(event.request)
          .then(response=>{
            if (response.status == 200) {
              caches.open(CACHE_GROUP).then(cache=>{
                cache.put(event.request, response.clone());
              });
            }
            return response;
          });
      });
  );
});
```

# Fetch from Cache

```
self.addEventListener('fetch', event=>{
  const url = new URL(event.request.url);
  const NOT_FOUND = '/my-404.html';

  event.respondWith(
    caches.match(event.request)
      .then(result=>{
        return result || fetch(event.request)
          .then(response=>{
            if (response.status == 200) {
              caches.open(CACHE_GROUP).then(cache=>{
                cache.put(event.request, response.clone());
              });
            } else {
              return caches.match(NOT_FOUND);
            }
          });
      });
  );
});
```

# IndexedDB

# SW tem acesso ao IndexedDB

- Pode ser usado para armazenar o resultado de requisições a JSONs
- Assíncrono (apesar da sintaxe "feia")
- Permite filtragem, pesquisa e atualizações

# ImportScripts



# Importing Scripts

```
importScripts('helper-script.js',  
             'indexeddb-abstraction.js');
```

# Web Notification

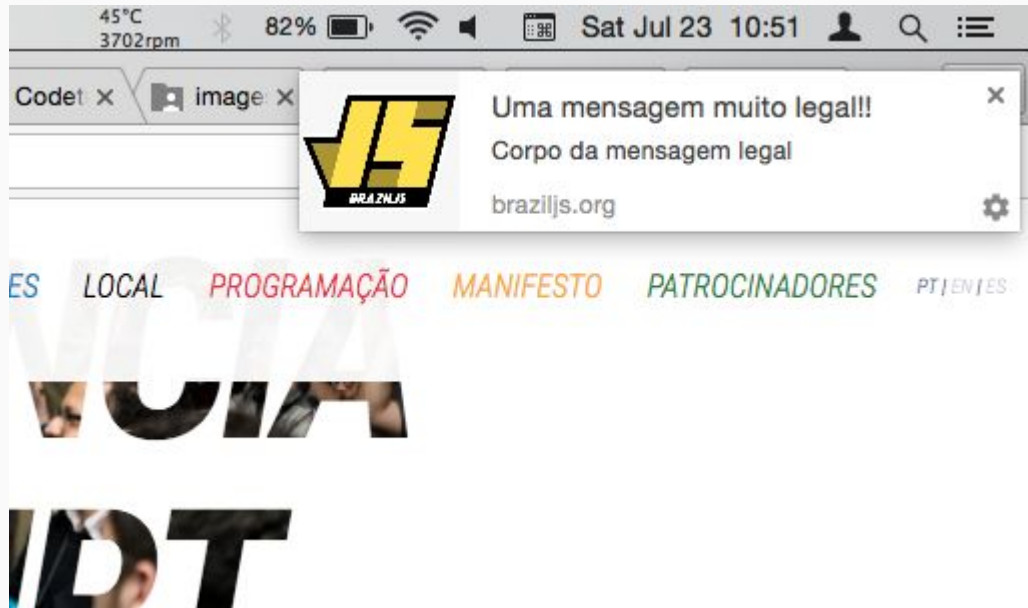
# Web Notification

```
Notification.requestPermission().then(function(result) {  
    console.log(result);  
});
```

# Web Notification

```
var options = {  
    body: theBody,  
    icon: theIcon  
}  
var n = new Notification(theTitle, options);  
setTimeout(n.close.bind(n), 5000);
```

# Web Notification



# Web Notification

webapp-manifest.json

```
"gcm_sender_id": "your-gcm-sender-id",
```

# Web Notification

Index.html (page script)

```
navigator.serviceWorker
  .ready
  .then(function(serviceWorkerRegistration) {
    serviceWorkerRegistration.pushManager.subscribe()
  })
);
```

# Web Notification

sw.js

```
self.addEventListener('push', function(event) {  
  console.log('Received a push message', event);  
  
  event.waitUntil(  
    self.registration.showNotification('A message!', {  
      body: "Message content",  
      icon: '/images/notif-icon.png',  
      tag: 'push-notification-test'  
    })  
  );  
});
```



MANIFEST

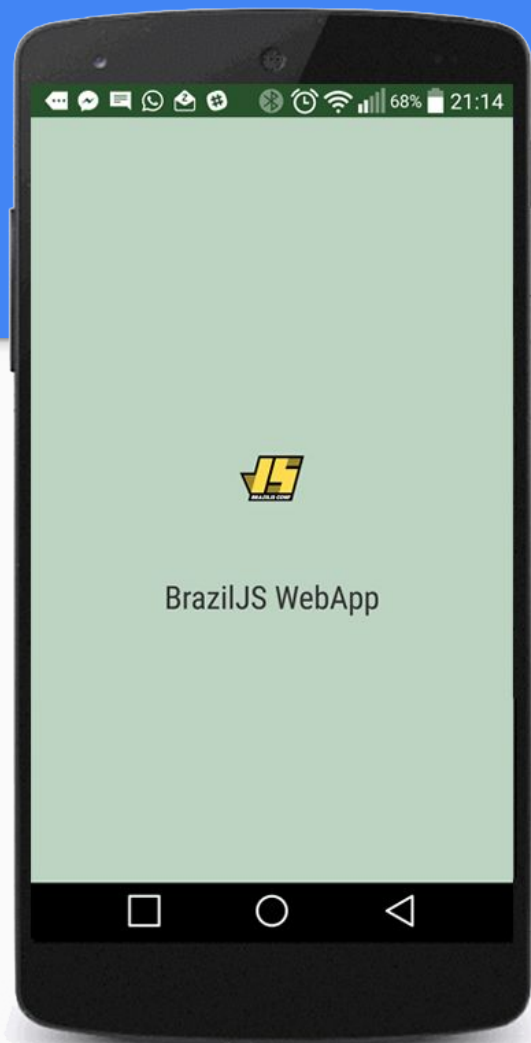
# index.html

```
<!-- WEB APP META DATA -->  
<link rel="manifest" href="/webapp-manifest.json">  
<meta name="theme-color" content="#438948">  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<!-- /WEB APP META DATA -->
```

# webapp-manifest.json

```
{  
  "name": "Your Web App Name",  
  "short_name": "appName",  
  "description": "Main description of your web app.",  
  "icons": [  
    {"src": "images/icon-128.png", "sizes": "128x128", "type": "image/png"},  
    {"src": "images/icon-152.png", "sizes": "152x152", "type": "image/png"},  
    {"src": "images/icon-144.png", "sizes": "144x144", "type": "image/png"},  
    {"src": "images/icon-192.png", "sizes": "192x192", "type": "image/png"}  
  ],  
  "start_url": "/index.html?homescreen=1",  
  "display": "standalone",  
  "background_color": "#47b74f",  
  "theme_color": "#438948"  
}
```

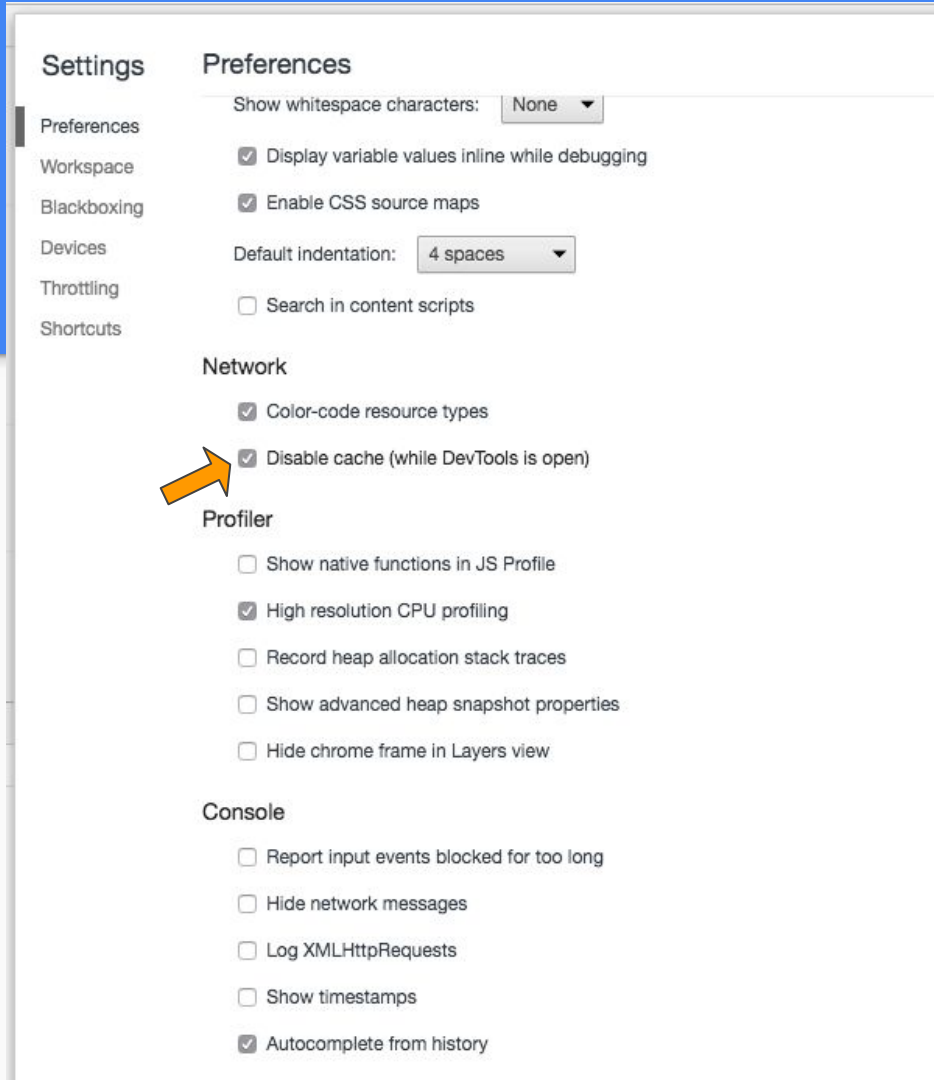




Hard times!

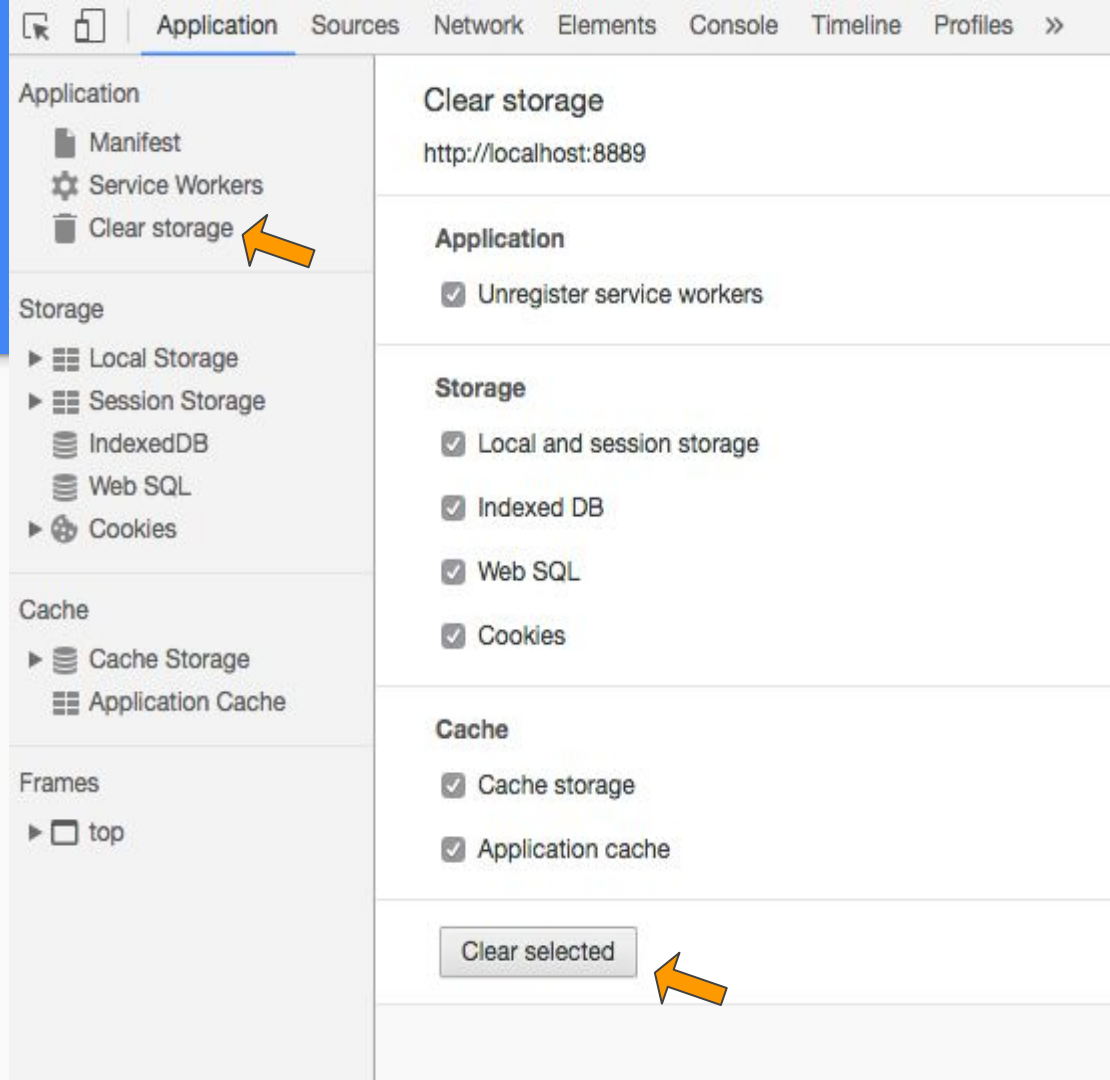
# Hard times!

- Desligue o cache  
(no console do browser)



# Hard times!

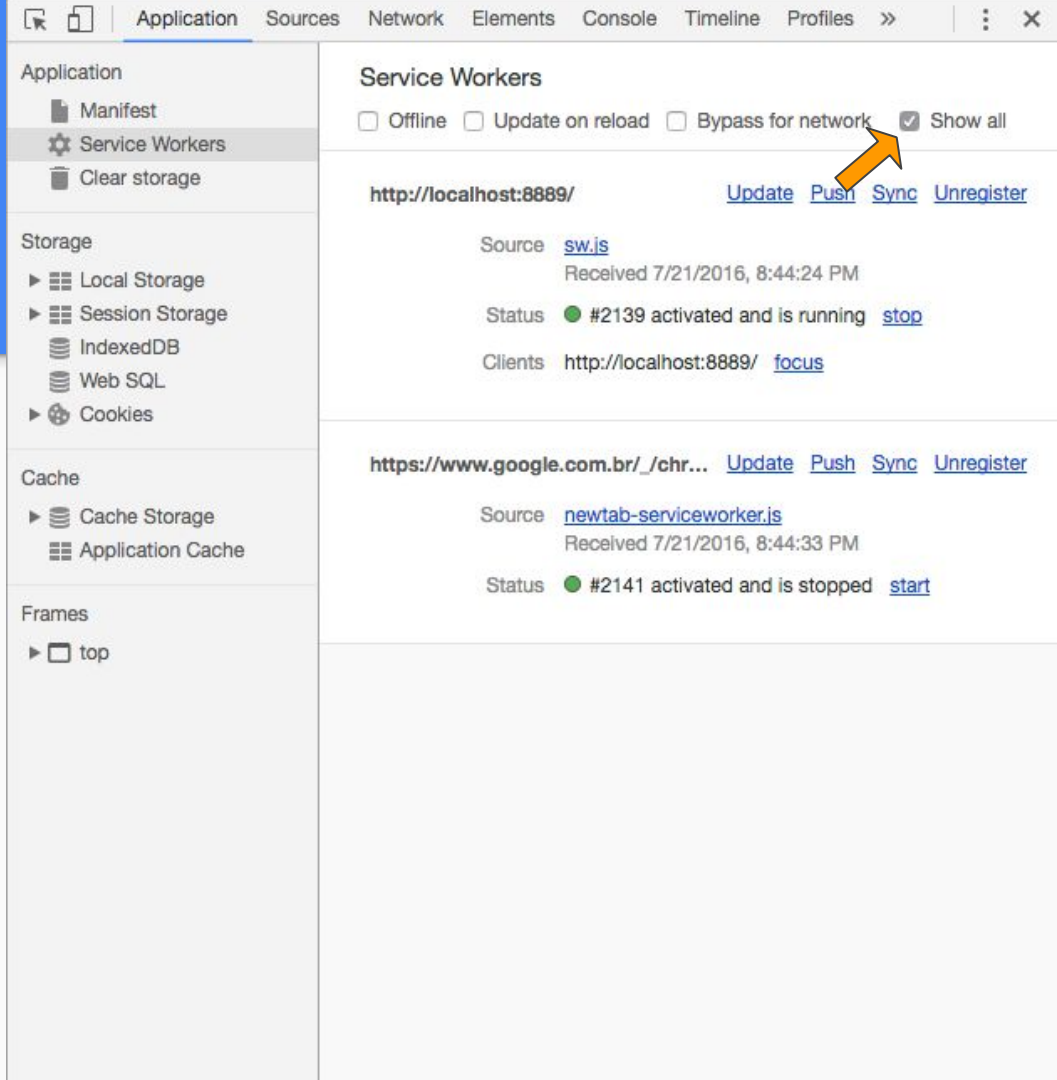
- Desligue o cache (no console do browser)
- Limpe os dados do cache





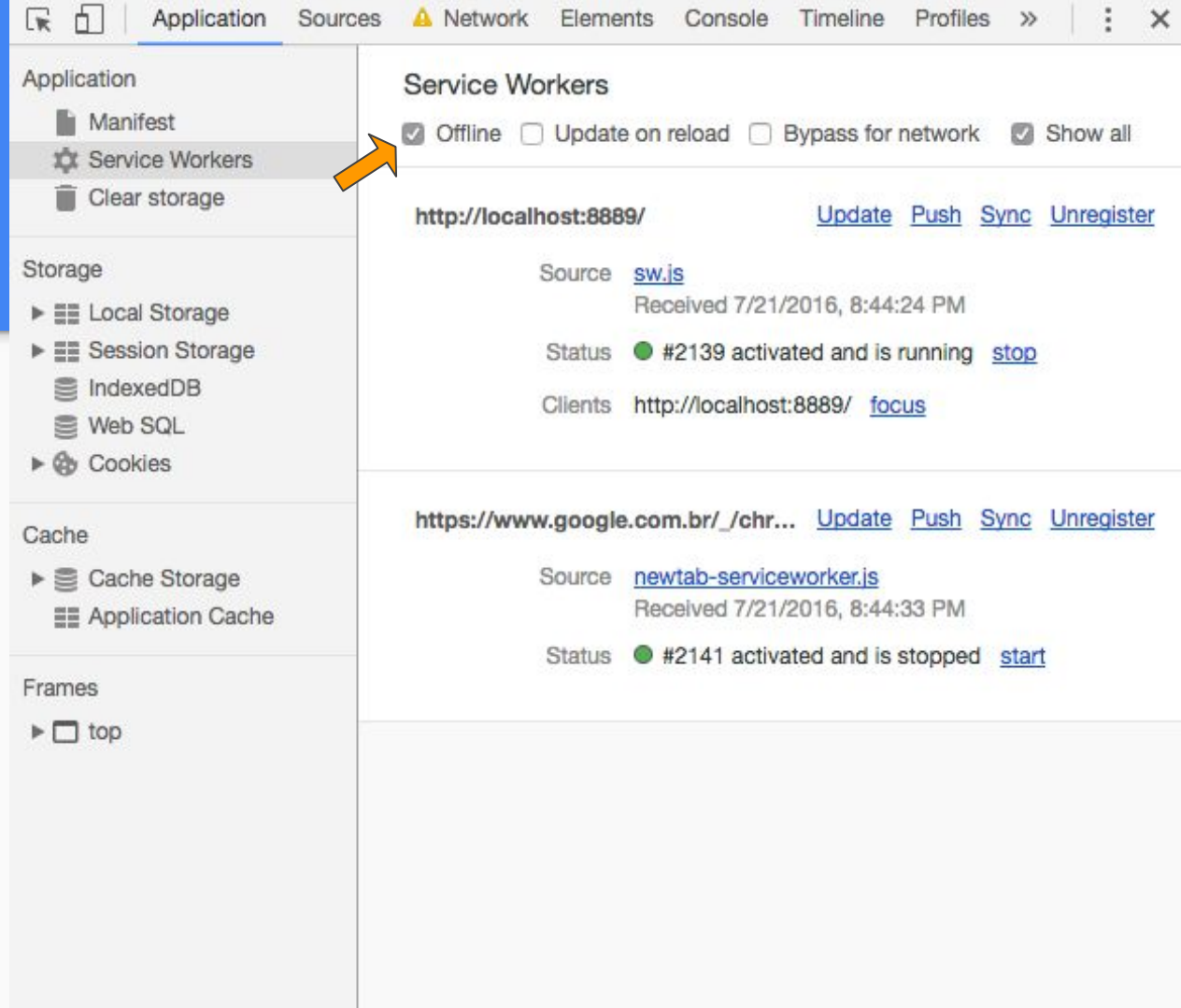
# Hard times!

- Desligue o cache  
(no console do browser)
- Limpe os dados do cache
- Visualize todos os  
Service workers



# Hard times!

- Teste offline



The screenshot shows the Chrome DevTools Application tab. The left sidebar has a tree view with categories: Application, Storage, Cache, and Frames. Under 'Application', 'Service Workers' is selected and highlighted with an orange arrow. The main panel displays the 'Service Workers' section for the current page. It includes a header with a checked 'Offline' checkbox and other options like 'Update on reload', 'Bypass for network', and 'Show all'. Below this, two service workers are listed:

Service Worker	Source	Received	Status	Actions
<a href="http://localhost:8889/">http://localhost:8889/</a>	<a href="#">sw.js</a>	7/21/2016, 8:44:24 PM	● #2139 activated and is running <a href="#">stop</a>	<a href="#">Update</a> <a href="#">Push</a> <a href="#">Sync</a> <a href="#">Unregister</a>
<a href="https://www.google.com.br/_chr...">https://www.google.com.br/_chr...</a>	<a href="#">newtab-serviceworker.js</a>	7/21/2016, 8:44:33 PM	● #2141 activated and is stopped <a href="#">start</a>	<a href="#">Update</a> <a href="#">Push</a> <a href="#">Sync</a> <a href="#">Unregister</a>

# Hard times!

- Teste offline
- Teste em lie-fi

The image shows two screenshots of the Chrome DevTools Network tab. The top screenshot shows the 'Network' tab selected, with a red dot icon indicating offline status. An orange arrow points to the 'Network' tab, and another orange arrow points to the 'Offline' checkbox, which is currently unchecked. The bottom screenshot shows the same interface, but with the 'Offline' checkbox checked. An orange arrow points to the 'Offline' checkbox, and another orange arrow points to the 'Presets' dropdown menu, which is open and showing various network simulation options.

Custom  
Add...

Presets

- Offline (0ms, 0kb/s, 0kb/s)
- GPRS (500ms, 50kb/s, 20kb/s)
- Regular 2G (300ms, 250kb/s, 50kb/s)
- Good 2G (150ms, 450kb/s, 150kb/s)
- Regular 3G (100ms, 750kb/s, 250kb/s)
- Good 3G (40ms, 1.5Mb/s, 750kb/s)
- Regular 4G (20ms, 4.0Mb/s, 3.0Mb/s)
- DSL (5ms, 2.0Mb/s, 1.0Mb/s)
- WiFi (2ms, 30Mb/s, 15Mb/s)

Disabled

No throttling

Name	Status	Type	Initiator	Size	Time	Timeline – Start Time▲

# Hard times!

- Teste offline
- Teste em lie-fi
- Use o lighthouse

Lighthouse

http://localhost:8889/

ALPHA

Progressive Web App

Best Practices

Performance Metrics

View: ☐ User feature ☒

94 / 100

These audits validate the aspects of a Progressive Web App.

App can load on offline/flaky connections

Ensuring your web app can respond when the network connection is unavailable or flaky is critical to providing your users a good experience. This is achieved through use of a [Service Worker](#).

Has a registered Service Worker

Yes

URL responds with a 200 when offline

Yes

Manifest's start\_url is in cache storage for offline use (Coming soon)

🔧

Page load performance is fast

Users notice if sites and apps don't perform well. These top-level metrics capture the most important perceived performance concerns.

First meaningful paint (target value: 1,000ms)

100 (173.94ms)

Speed Index (target value: 1,000)

100

Content scrolls at 60fps (Coming soon)

🔧

Touch input gets a response in < 150ms (Coming soon)

🔧

App is interactive without jank after the first meaningful paint (Coming soon)

🔧

Site is progressively enhanced

Progressive enhancement means that everyone can access the basic content and functionality of a page in any browser, and those without certain browser features may receive a reduced but not unfunctioning experience.

Page contains some content when its scripts are not available

Yes

Network connection is secure

Security is an important part of the web for both developers and users. Moving forward, Transport Layer Security (TLS) support will be required for many APIs.

Site is on HTTPS

Yes

TOOLS

# Meet the DSW

The

Dynamic  
Service  
Worker

## DSW Try it Yourself page

Use this page to test and see it working.

Open your browser's console and check the cached content, and the network requests as you test the features.

### Load image

This example will successfully load an image from cache.  
This image loads dynamically and will be cached only after the first time it loaded successfully.

► Run test

01

02

### Image not found

This tries to load an image that does not exist.  
It loads then, a default 404 image.  
The 404 image was cached during the Service Worker installation.

► Run test

### Redirected Image

03

# Meet the DSW

```
$ npm install -g dsw
```

GitHub, Inc. [US] <https://github.com/NaschHQ/dsw>



This repository

Search

NaschHQ / dsw

## Dynamic Service Worker



Version v1.3.2

codacy A

license MIT

issues 6 open

build passing

DSW allows you to enable and use Service Workers in a much easier way, also helping Progressive Web Apps working offline.

You will simply have to create setup in a JSON file how your service worker is supposed to work. You can find a [commented JSON example](#) or the documentation and examples below.

# DSW - import and call .setup

```
<link rel="manifest" href="/webapp-manifest.json" />
<meta name="theme-color" content="#color">
<script src="dsw.js"></script>
<script>
  DSW.setup()
    .then(function(){
      // inform the user your page works offline, now!
    })
    .catch(function(){
      // do something if the page will not work offline
      // or if the current browser does not support it
    });
</script>
```



# dswfile.json

```
{  
  "dswVersion": 2.2,  
  "applyImmediately": true,  
  "appShell": [],  
  "enforceSSL": false,  
  "dswRules": {  
    "myRule": {  
      "match": { /* path or extension */ },  
      "apply": { /* fetch, cache, redirect or idb */ }  
    }  
  }  
}
```

# DSW

```
felipenmoura :: /tests/sw $  
felipenmoura :: /tests/sw $ dsw  
[DSW] :: Tip ::  
    Don't forget to link your manifest in your index.html file:  
    <link rel="manifest" href="/webapp-manifest.json">  
    <meta name="theme-color" content="#color">  
[DSW] :: Start your HTTP server at:  
    /tests/sw/  
[DSW] :: Done, now go play outside!  
felipenmoura :: /tests/sw $
```

# DSW - Not found HTML

```
{
  "dswVersion": 2.2,
  "dswRules": {
    "notFoundPages": {
      "match": {
        "status": [404],
        "extension": ["html"]
      },
      "apply": {
        "fetch": "/my-404-page.html"
      }
    }
  }
}
```

# DSW - not found images

```
"imageNotFound": {  
  "match": {  
    "status": [404, 500],  
    "extension": ["jpg", "gif", "png", "jpeg", "webp"]  
  },  
  "apply": {  
    "fetch": "/images/public/404.jpg"  
  }  
},
```

# DSW - caching images

```
{
  "dswVersion": 2.2,
  "dswRules": {
    "myCachedImages": {
      "match": {
        "extension": ["png", "jpg", "gif"]
      },
      "apply": {
        "cache": {
          "name": "my-cached-images",
          "version": 1
        }
      }
    }
  }
}
```

# DSW - Redirecting

```
{
  "dswVersion": 2.2,
  "dswRules": {
    "secretPath": {
      "match": {
        "path": "\/private\/"
      },
      "apply": {
        "redirect": "/not-allowed.html"
      }
    }
  }
}
```

# Redirect com variáveis

```
{
  "dswVersion": 2.2,
  "applyImmediately": true,
  "appShell": [],
  "enforceSSL": false,
  "dswRules": {
    "redirectWithVar": {
      "match": {
        "path": "\/old-site\/(.*)"
      },
      "apply": {
        "redirect": "\/redirected.html?from=$1"
      }
    }
  }
}
```

# DSW - caching css and js

```
{
  "dswVersion": 2.2,
  "dswRules": {
    "statics": {
      "match": {
        "extension": ["js", "css"]
      },
      "apply": {
        "cache": {
          "name": "page-static-files"
          "version": 1
        }
      }
    }
  }
}
```



# Outras ferramentas

## SW-Toolbox

The screenshot shows the GitHub repository page for `GoogleChrome/sw-toolbox`. The browser address bar displays the URL `https://github.com/GoogleChrome/sw-toolbox`. The repository name is `GoogleChrome / sw-toolbox`. The page includes navigation links for `Pull requests`, `Issues`, and `Gist`. Below the repository name, there are tabs for `Code`, `Issues 34`, `Pull requests 3`, `Wiki`, `Pulse`, and `Graphs`. The description of the repository is "A collection of service worker tools for runtime requests" with a link to `https://googlechrome.github.io/sw-toolbox/`. The repository statistics show `225 commits`, `7 branches`, and `13 releases`. At the bottom, there is a commit by `wibblymat` titled "Fixes #157" with a list of files changed: `doc-pages` (Fixing short link) and `lib` (Fixes #3).

GitHub, Inc. [US] <https://github.com/GoogleChrome/sw-toolbox>

This repository Search Pull requests Issues Gist

GoogleChrome / **sw-toolbox** Watch

Code Issues 34 Pull requests 3 Wiki Pulse Graphs

A collection of service worker tools for runtime requests <https://googlechrome.github.io/sw-toolbox/>

225 commits 7 branches 13 releases

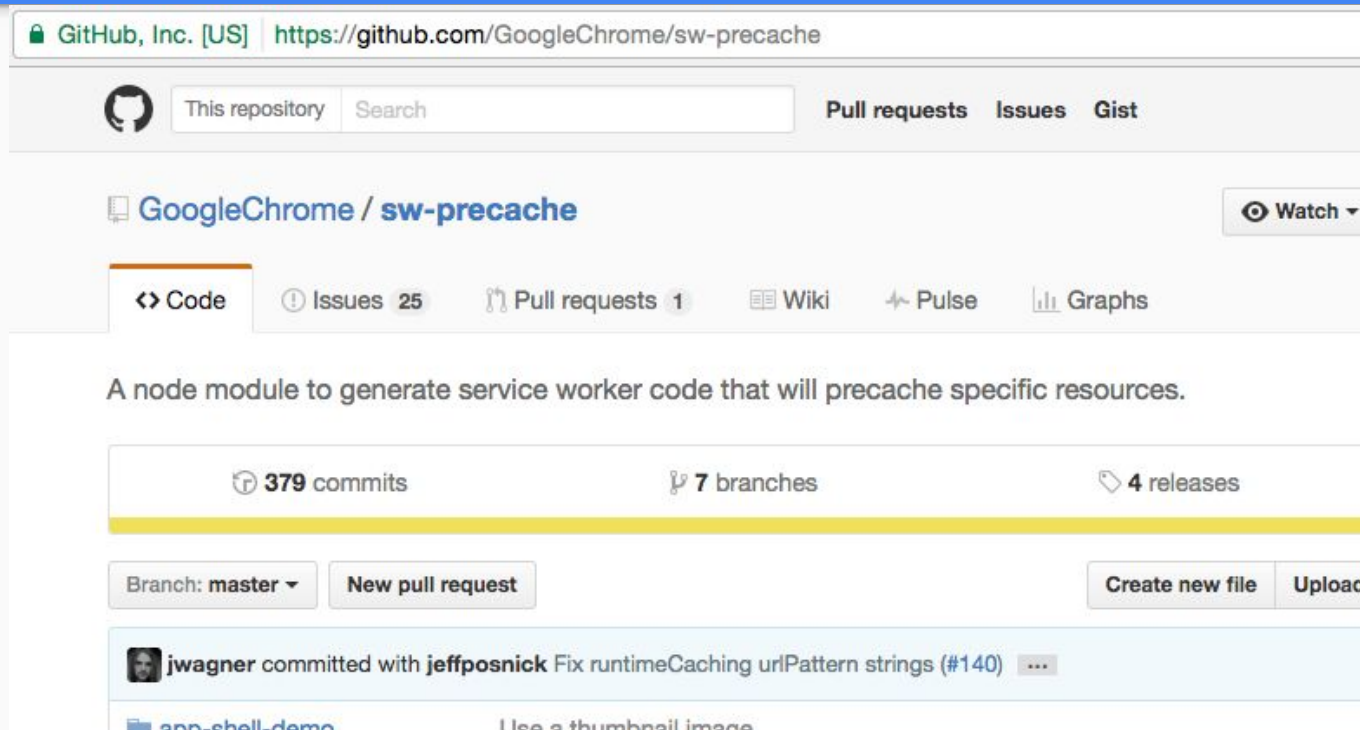
Branch: master New pull request Create new file Upload file

wibblymat committed on GitHub Fixes #157

doc-pages	Fixing short link
lib	Fixes #3

# Outras ferramentas

## SW-Precache



The screenshot shows the GitHub repository page for `GoogleChrome/sw-precache`. The page has a blue header with the repository name and a search bar. Below the header, there are tabs for `Code`, `Issues` (25), `Pull requests` (1), `Wiki`, `Pulse`, and `Graphs`. The description of the repository is "A node module to generate service worker code that will precache specific resources." Below the description, there are statistics: 379 commits, 7 branches, and 4 releases. At the bottom, there is a commit by `jwagner` with `jeffposnick` as the co-author, titled "Fix runtimeCaching urlPattern strings (#140)".

GitHub, Inc. [US] <https://github.com/GoogleChrome/sw-precache>

This repository Search Pull requests Issues Gist


GoogleChrome / **sw-precache** Watch

<> Code ! Issues 25 🔗 Pull requests 1 📖 Wiki ⚡ Pulse 📊 Graphs

A node module to generate service worker code that will precache specific resources.

🕒 379 commits 🌿 7 branches 📦 4 releases

Branch: master ▼ New pull request Create new file Upload

 **jwagner** committed with **jeffposnick** Fix runtimeCaching urlPattern strings (#140) ...

📁 app-shell-demo Use a thumbnail image

TKS!

Felipe N. Moura



# Thank you!

@felipeNmoura

(github || facebook) / felipenmoura

[felipenmoura.com](http://felipenmoura.com)

