FUTXICAIADA

FUTXICAIADA

WEB WORKER – PROMESSAS

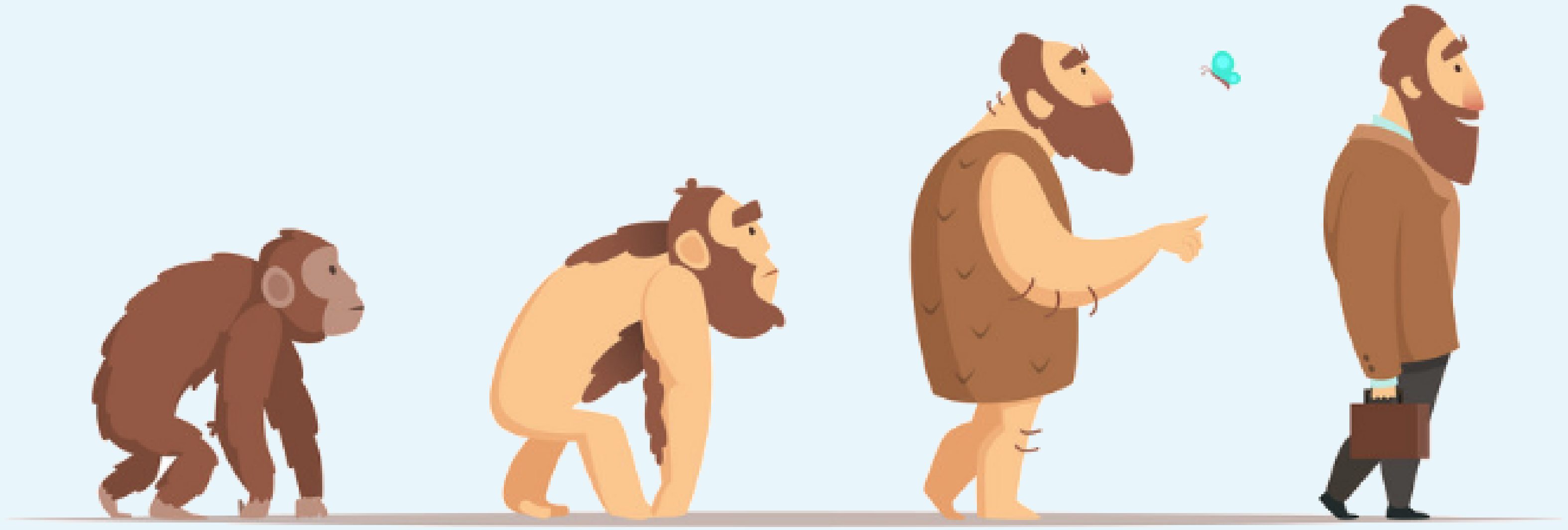
Itacir Gabral

🙋🏽‍♂️ Pode perguntar!

FUTXICAIADA

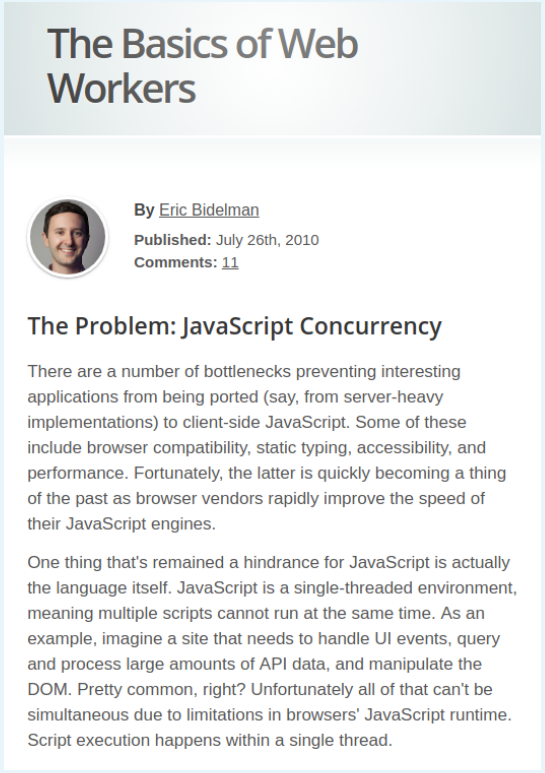
http://dontpad.com/futx

WEB WORKER



Web Worker – – – – – – – – – > Service Worker

WEB WORKER



The Basics of Web Workers

By Eric Bidelman

Published: July 26th, 2010

Comments: 11

The Problem: JavaScript Concurrency

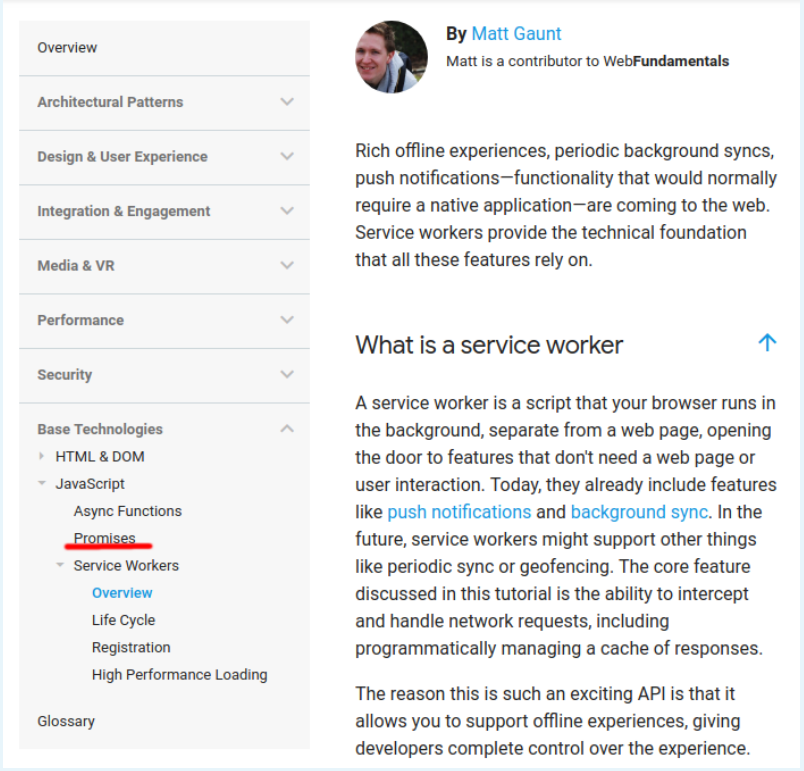
There are a number of bottlenecks preventing interesting applications from being ported (say, from server-heavy implementations) to client-side JavaScript. Some of these include browser compatibility, static typing, accessibility, and performance. Fortunately, the latter is quickly becoming a thing of the past as browser vendors rapidly improve the speed of their JavaScript engines.

One thing that's remained a hindrance for JavaScript is actually the language itself. JavaScript is a single-threaded environment, meaning multiple scripts cannot run at the same time. As an example, imagine a site that needs to handle UI events, query and process large amounts of API data, and manipulate the DOM. Pretty common, right? Unfortunately all of that can't be simultaneous due to limitations in browsers' JavaScript runtime. Script execution happens within a single thread.

html5rocks.com

26/jul/2010

WEB WORKER



By Matt Gaunt

Matt is a contributor to Web**Fundamentals**

Rich offline experiences, periodic background syncs, push notifications—functionality that would normally require a native application—are coming to the web. Service workers provide the technical foundation that all these features rely on.

What is a service worker

A service worker is a script that your browser runs in the background, separate from a web page, opening the door to features that don't need a web page or user interaction. Today, they already include features like push notifications and background sync. In the future, service workers might support other things like periodic sync or geofencing. The core feature discussed in this tutorial is the ability to intercept and handle network requests, including programmatically managing a cache of responses.

The reason this is such an exciting API is that it allows you to support offline experiences, giving developers complete control over the experience.

WebFundamentals

WEB WORKER

- Dedicados
- Compartilhados
- Service Workers
- Worklets

PROMESSAS

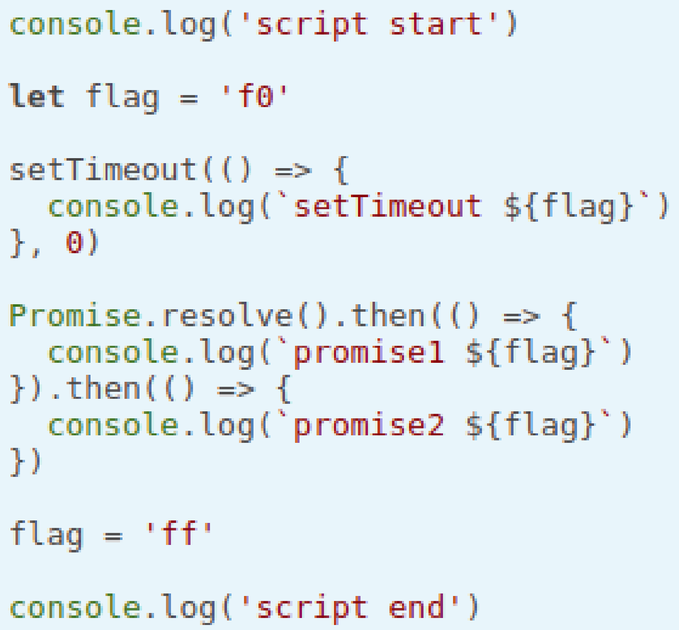
BlueBird*

4x mais rápido

4x menos memória

* 2016. Em comparação com promessas nativas.

PROMESSAS



```
console.log('script start')

let flag = 'f0'

setTimeout(() => {
  console.log(`setTimeout ${flag}`)
}, 0)

Promise.resolve().then(() => {
  console.log(`promise1 ${flag}`)
}).then(() => {
  console.log(`promise2 ${flag}`)
})

flag = 'ff'

console.log('script end')
```

Fila de eventos

Fila de trabalhos

Fila de micro-tarefas

REVISÃO

```
var somaDois = function somaDois (n) {
  return n + 2
}
```

```
const somaDois = n => n + 2
```

REVISÃO

```
var soma = function soma (k) {
  return function somaK (n) {
    return n + k
  }
}
```

```
const soma = k => n => n + k
```