

02

Atualizando nossa Web App - Parte 2

Transcrição

Já podemos começar a alterar os arquivos, atualizando a app sempre que quisermos. Como exemplo, vamos modificar a cor de fundo novamente, dessa vez para verde (*green*, em inglês). Para que isto funcione no navegador, atualizarei a versão tanto no Application Cache (abrindo `offline.manifest` e trocando `#v3` por `#v4`) quanto no Service Worker (em `service-worker.js`, `let versao = 3` vira `let versao = 4`).

Com tudo atualizado, recarregarei a página, a qual mostra que o Service Worker foi instalado, pois vemos através da aba "Application" que a numeração do Service Worker novo é outra. O nosso cabeçalho ficará verde na próxima vez em que o arquivo `cabecalho.css` for acessado. Para demonstrar isto, recarregarei a página, e o cabeçalho ficará verde:

The screenshot shows the Chrome DevTools Application tab for the 'Ceep' application. The sidebar on the left has sections for Application (Manifest, Service Workers, Clear storage), Storage (Local Storage, Session Storage, IndexedDB, Web SQL), and Network. The main area shows the Service Workers for the URL `http://localhost:3000/`. The service worker source is `service-worker.js`, received on 2/6/2017, 12:57:22 AM. The status is `#190 activated and is running`, with a link to `stop`. There is also an entry for `#222 waiting to activate` from 2/6/2017, 3:31:15 AM, with a link to `skipWaiting`. The status bar at the bottom of the DevTools is highlighted in yellow.

Abrirei outra aba com a url [localhost:3000 \(http://localhost:3000\)](http://localhost:3000). Trata-se da mesma aplicação em duas abas diferentes, e os arquivos utilizados são os mesmos também. Atualizarei a app alterando a cor de fundo do cabeçalho mais uma vez, passando agora para roxo (*purple*). Teremos que efetuar todas aquelas alterações nos números das versões, como visto anteriormente (a versão atual, no caso, é a 5).

A primeira aba recarregada demonstra que o número do Service Worker instalado é outro, o que significa que ele foi instalado, ou seja, da próxima vez o cabeçalho já vai estar na cor roxa. A segunda aba, que já estava aberta enquanto fazíamos a alteração na anterior, continua na versão antiga. Este é o comportamento esperado. Se não for atualizado, ele precisa permanecer com o que já possuía.

No entanto, quando recarreguei uma das abas, a outra também teve um novo Service Worker atualizado: são duas versões concomitantes. Em se tratando de uma cor de cabeçalho, isto não significa muita coisa, mas imagine que tivessemos

adicionado novas funcionalidades aos cartões, corrigido um *bug*, ou mexido em algo que "quebre" uma versão da outra...

Este código estaria sendo executado somente na versão mais recente, e não mais na antiga. Podemos evitar que isto aconteça. Como vimos, o Cache Storage é atualizado sempre que um novo Service Worker é baixado. Atualizamos o cache no evento de `install` e esta instalação acontece sempre que o `service-worker.js` passa por alguma alteração.

Se quisessemos garantir que todas as abas sejam atualizadas ao mesmo tempo, o que seria garantido, na verdade, é que a app seja atualizada quando o Service Worker for ativado. É exatamente o que ainda não ocorreu neste caso. Houve instalação, e consequente visualização da alteração da cor de cabeçalho, porém sua ativação ainda não foi efetuada.

O Service Worker é ativado quando todas as abas são fechadas, quando todos os próximos recarregamentos estarão automaticamente com a versão nova. Só depois que todas as abas forem fechadas e não existir mais ninguém utilizando a versão antiga é que o Service Worker é ativado. Para assegurar que a app esteja sempre na mesma versão mesmo em abas diferentes, precisamos que este código, o qual atualiza o cache, não seja executado em `install`, e sim quando o Service Worker for ativado.

Vamos alterar o código que modifica o que tínhamos antes, e que atualiza a aplicação, colocando-o dentro de `self.addEventListener`, cujo evento ouvido não deixará de ser o de instalação, e passará a ser de **activate** (ativação). Todo Service Worker dispara este evento em algum momento, por padrão. Então, quando nenhuma aba estiver aberta na versão antiga, é que aproveitaremos para usar a versão mais recente.

```
self.addEventListener("install", function(){
  console.log("Instalou")
})

self.addEventListener("activate", function(){
  caches.open("ceep-arquivos-" + versao).then(cache => {
    cache.addAll(arquivos)
    .then(function(){
      caches.delete("ceep-arquivos-" + (versao - 1))
      caches.delete("ceep-arquivos")
    })
  })
})
```

Testaremos atualizando a versão (de 5 para 6, neste caso), nos arquivos `offline.manifest` e `service-worker.js`. Alteraremos também a cor de fundo do cabeçalho novamente, para amarelo (*yellow*). Agora, ao recarregarmos a página, a instalação do Service Worker será realizada. Até então, o cache estava sendo atualizado, porém ele não existe mais, seja lá qual for a versão do Service Worker instalada em cada aba aberta. O importante é que esta versão só será atualizada quando nenhuma aba estiver aberta.

Quando abrimos uma página para carregarmos o [servidor \(http://localhost:3000\)](http://localhost:3000), é a versão nova e ativa que nos é mostrada:

The screenshot shows the Chrome DevTools interface with the Application tab selected. On the left, a sidebar lists 'Application', 'Manifest', 'Service Workers' (which is selected and highlighted in blue), and 'Clear storage'. Under 'Storage', there are links for 'Local Storage', 'Session Storage', 'IndexedDB', and 'Web SQL'. The main content area is titled 'Service Workers' and shows a list for the URL <http://localhost:3000/>. It displays the source file as 'service-worker.js', the last update time as 'Received 2/6/2017, 3:36:45 AM', and the status as '#234 activated and is running'. A 'stop' link is provided for stopping the worker. Below this, a 'Clients' section lists the URL <http://localhost:3000/> and a 'focus' link.

Portanto, os Service Workers são bem poderosos, não se tratando apenas de eventos de instalação que podem ser ouvidos. Através dos eventos de ativação, podemos atualizar as apps de maneira mais consistente.