

Pegando todas as respostas do Service Worker

Transcrição

Vamos verificar se tudo está funcionando no navegador. O Service Worker ativo ainda é o antigo (#121), portanto vou recarregar a página para que o Service Worker novo (#123) seja percebido, sendo baixado e pronto para ser ativado. Então, neste momento temos dois Service Workers rodando: o antigo, sendo utilizado no momento, e outro que ainda não foi ativado, porém está pronto para uso. Em resumo, o Service Worker funciona a partir do momento em que é ativado, o que acontece automaticamente, através do navegador.

Poderíamos fechar todas as abas abertas na aplicação, como já feito anteriormente, e o Service Worker seria ativado, já que ele funciona da mesma forma para todo mundo.

No Developer Tools, ao clicarmos em "**skipWaiting**" ao lado da informação do Service Worker novo, o ativamos, e ele está pronto para uso. Quando a página é recarregada, a imagem não vem mais do servidor, e sim do Cache Storage.

Vou à aba "Network", limpo tudo e recarrego a página. Ao fazer isto, aparece um erro:

Erro Network

Vê-se que [a página principal \(http://localhost:3000/\)](http://localhost:3000/) está tentando ser carregada, sem sucesso. Pode parecer estranho, mas o que aconteceu é que nós falamos para o site ser quebrado.

No Service Worker, para qualquer arquivo, sendo imagem ou não, inclusive para o arquivo principal do site, [localhost:3000 \(http://localhost:3000/\)](http://localhost:3000/), indicamos ao navegador que se deve procurar todos os arquivos no cache, sempre. O que acontece é que até mesmo os arquivos `.css`, `.js` e o `index.html` são procurados no Cache Storage. Quando eles não são encontrados, a resposta vem vazia, sem nada dentro, o que quer dizer que só podemos utilizá-la de verdade caso ela encontre algo no cache, uma imagem, por exemplo.

Nem sempre será possível retornar essa resposta. Quando não houver uma, teremos que utilizar os dados disponíveis no servidor, e usar a resposta dali, não do cache. Neste caso, posso executar um código que possibilite o uso da resposta vinda do servidor. Ou seja, a resposta pode vir do cache ou do servidor.

A variável `respostaCache` corresponde a uma resposta que não será sempre a mesma. Caso não haja uma resposta proveniente do cache, verificarei se isto é verdadeiro ou falso, se está vazio ou não. Se houver resposta do cache, vou usá-la e, caso contrário, solicitarei ao servidor. Estarei utilizando um "**if ternário**". Com um request pronto, posso usar a função `fetch()`, disponível no Service Worker, que permite um request assíncrono ao servidor, para o qual passarei o pedido completo sem precisar ter uma url específica, algo que o browser já faria ao servidor. Como retorno, posso passar a função `fetch()`, pois é uma promise cujo resultado é uma resposta. O código abaixo (do arquivo `service-worker.js`) permite uma resposta válida para a `event.respondWith`:

```
self.addEventListener("fetch", function(event){

  let pedido = event.request
  let promiseResposta = caches.open("ceep-imagens").then(cache => {
    return cache.match(pedido)
  }).then(respostaCache) => {
    let resposta = respostaCache ? respostaCache : fetch(pedido)
    return resposta
  }
```

```
})
```

```
event.respondWith(promiseResposta)
```

```
})
```

Obtemos uma resposta proveniente do cache, ou através da `fetch()`, que acessa o servidor. Salvarei este arquivo, recarregando a página em seguida. Não tenho mais acesso ao Service Workers, não sei o que está acontecendo ali, o site não está nem conseguindo ser acessado. Para verificar isto, na aba "Application" existe a opção "Show all" (mostrar tudo), e isto me mostra todos os Service Workers cadastrados no navegador, como o do YouTube, por exemplo. Um deles é o da nossa página, [localhost:3000 \(http://localhost:3000/\)](http://localhost:3000/).

Recarregarei a página para checar a funcionalidade do Service Worker, que precisa ser atualizado. Aparentemente, está tudo funcionando corretamente. Vamos conferir de onde estão vindo os arquivos a partir da aba "Network":

Origem arquivos

Eles vêm do Service Worker, como se vê. Vamos ver os outros arquivos? Todos eles também vêm do mesmo lugar:

Origem arquivos

Isso é verdade pois não significa que eles vêm necessariamente do cache, mas todos os arquivos estão sendo servidos por ele ou pelo `fetch()`, e é para elucidar isso que confirmamos sua origem dos Service Workers. No entanto, como temos certeza de que a imagem que vemos vem realmente do Cache Storage, e não do servidor, do GitHub? Se a imagem está sendo acessada pelo Application Cache, isto deveria funcionar offline. Vamos fazer este teste, marcando a opção "Offline" e recarregando a página. Nada está funcionando!

Como faço para verificar os erros dentro dos Service Workers? Vou ao console, cliço no erro, o que permite que vejamos sua mensagem que, neste caso, diz que não foi possível baixar o Service Worker. Houve outros erros também, como não conseguir baixar a url [localhost:3000 \(http://localhost:3000/\)](http://localhost:3000/).

Os arquivos vinham do Application Cache e, no momento em que um Service Worker é adicionado à sua página, o mesmo Application Cache é eliminado. Trata-se de uma versão antiga das apps dizerem que funcionavam offline. O Service Workers, assim como as Cache Storages, vieram para substituir o Application Cache. Então, quando ativamos os Service Workers, acabamos perdendo o Application Cache. Caso queiramos seguir utilizando o Service Workers e o Cache Storage para acessar suas imagens, também teremos que usar o Cache Storage para acessar todos os arquivos do site. Precisaremos substituir completamente o Application Cache pelo Cache Storage. É esse o código que iremos implementar em seguida.

