

JAVASCRIPT ASSÍNCRONO

History API

History

É possível acessarmos o histórico de acesso do browser em uma sessão específica através do `window.history`. O objeto history possui diferentes métodos e propriedades.

```
window.history;  
window.history.back(); // vai para a anterior  
window.history.forward(); // vai para a próxima
```

pushState()

A parte interessante de manipularmos o history é que podemos modificar o histórico e adicionar um novo item.

```
window.history.pushState(obj, title, url).
```

```
// Em obj podemos enviar um objeto com dados  
// mas o seu uso é restrito por isso geralmente utilizamos  
// null. O title ainda é ignorado por alguns browsers, também  
// utilizamos null nele. O url que é parte importante.
```

```
window.history.pushState(null, null, 'sobre.html');
```

popstate

O evento `popstate` pode ser adicionado ao objeto `window`. Assim podemos executar uma função toda vez que o usuário clicar no botão de voltar ou próximo.

```
window.addEventListener('popstate', () => {  
  fetchPage(window.location.pathname);  
});
```

[COPIAR](#)

Fetch e History

Ao puxarmos dados via fetch api, o url da página continua o mesmo. Ao combinar fetch com a history api conseguimos simular uma navegação real entre páginas, sem a necessidade de recarregamento da mesma.

```
async function fetchPage(url) {  
  const pageReponse = await fetch(url);  
  const pageText = await pageReponse.text();  
  window.history.pushState(null, null, url);  
}
```