



Diferenças entre Node.js e o navegador

Ambos browser e Node.js utilizam JavaScript como sua linguagem de programação.

Construir aplicações que rodem no browser é uma coisa completamente diferente de construir aplicações Node.js

Apesar do fato que é sempre JavaScript, há fatores chave que tornam a experiência radicalmente diferente.

Da perspectiva de um desenvolvedor frontend que usa JavaScript extensivamente, aplicações Node.js trazem consigo uma enorme vantagem: o conforto de programar tudo - o frontend e o backend - em uma única linguagem.

Você tem uma grande oportunidade porque nós sabemos quão difícil é para aprender, completa e profundamente, uma nova linguagem de programação, e por usar a mesma linguagem para fazer todo o trabalho na web - tanto no servidor quanto no cliente, você está em uma posição única de vantagem.

O que muda é o ecossistema.

No browser, na maioria do tempo o que você está fazendo é interagindo com o DOM, ou outras APIs Web como Cookies. Isso não existe no Node.js, é claro. Você não tem o `document`, `window` e todos os outros objetos que são providos pelo browser.

E no browser, nós não temos as APIs legais que o Node.js provê com

seus módulos, como a funcionalidade de acesso ao filesystem.

Outra grande diferença é que no Node.js você controla seu ambiente. A não ser que você esteja criando uma aplicação open source que qualquer um pode hospedar em qualquer lugar, você sabe em qual versão do Node.js a aplicação vai rodar. Comparado ao ambiente do browser, onde você não tem o luxo de escolher qual browser seu visitante vai utilizar, isso é muito conveniente.

Isso significa que você pode escrever códigos com os modernos ES6-7-8-9, se atentando ao suporte da sua versão do Node.js.

Visto que o JavaScript se move muito rápido, mas os browsers podem ser um pouco lentos para atualizarem, as vezes na web, você está preso em versões velhas do JavaScript / ECMAScript.

Você pode utilizar o Babel para transformar seu código em um formato compatível com ES5 antes de enviar para o browser, mas no Node.js, você não precisa disso.

Outra diferença é que o Node.js utiliza o sistema de módulos CommonJS, enquanto que nos browsers ainda estamos vendo o inicio da implementação do padrão ES Modules.

Na prática, isso signifca que por enquanto você utiliza `require()` no Node.js e `import` no browser.

[←](#)

Previous

Quanto de JavaScript você precisa saber para usar Node.js?

Next

O V8 JavaScript Engine

[→](#)