

O fantasma da incompatibilidade

Transcrição

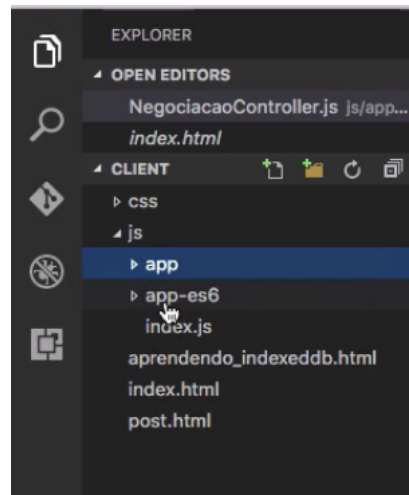
Começando deste ponto? Você pode fazer o [DOWNLOAD \(https://github.com/alura-cursos/javascript-avancado-iii/archive/aula5.zip\)](https://github.com/alura-cursos/javascript-avancado-iii/archive/aula5.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

O código que estamos escrevendo usa o que tem de mais moderno na linguagem JavaScript - já vimos o Fetch API, que é um assunto mais avançado. Mas o que fizemos até agora será suportado por navegadores que não aceitam ES2015? Não, porque eles não suportam `class`, `constructor` ou `let` ... Por exemplo, alguns celulares não rodam algumas destas funcionalidades. Mas devemos nos abdicar de escrever um código mais sucinto, por isso? Não. Pensaremos em uma solução.

Nós programaremos com o ES6 e depois, vamos compilar o código para o ES5. Este processo de *downgrade* recebe o nome de **transcompilação** e é feito com o uso de um *transpiler* (transcompilador). Com isto, o código da `Controller` consegue ter o mesmo resultado no ES 5. Desta forma, conseguimos aumentar a quantidade de navegadores que suportarão o nosso código, e assim, com a sua compatibilidade.

Encontramos vários *transpilers* no mercado: **Babel**, o **TypeScript** (que também atua como transcompilador). No nosso caso, focaremos no [Babel \(https://babeljs.io/\)](https://babeljs.io/), por ser open source.

A seguir, executaremos um passo **importante** no projeto: renomearemos a pasta `app` para `app-es6`. Com esta mudança, não precisaremos mudar o caminho dos `script` s. O objetivo é que ao alterarmos um arquivo, ele será automaticamente compilado em ES5 e serão colocados em um nova pasta `app`.



O código-fonte estará escrito com o ES 6, mas o resultado da transcompilação ficará na pasta `app`. E será desta que iremos importar os arquivos do `index.html`. Porém, se tentarmos recarregar a página do formulário agora, ela não funcionará porque nenhum `script` será encontrado. Veremos como fazer a transcompilação.