

05

Revisão: transpiler

Que tal reforçar ainda mais o que aprendeu sobre transpiler? Vem comigo!

Revisão

Durante todo o treinamento usufruimos como desenvolvedores dos recursos do ES2015 visando a escrita de um código mais elegante e mais fácil de manter. Contudo, não é raro o próprio desenvolvedor se questionar sobre a compatibilidade do seu código em relação aos seus usuários ou visitantes do site.

Normalmente é realizado um estudo (métricas do Google Analytics, por exemplo) para saber a predominância de determinados browsers para então deixar de suportar aqueles com uso mais tímido. Ainda assim, se algum browser com baixo suporte ao ES2015 for um dos mais usados alguma atitude deve ser tomada.

Dentro desse contexto, o desenvolvedor tem que se equilibrar na balança que ora pesa para o lado do que há de mais moderno da linguagem e ora para a questão de compatibilidade, esta última, vencedora justa a maior parte do tempo.

Para solucionar os problemas de compatibilidade e ainda permitir que o desenvolvedor utilize o que há de mais moderno da linguagem JavaScript foram criados **compiladores de código fonte para código fonte** comumente chamados de transcompiladores (transpilers). Com eles, é possível, por exemplo, converter um código fonte de Ruby para JavaScript e vice-versa. No entanto, no universo JavaScript a ideia é compilarmos um código-fonte escrito em ES2015 para ES5, garantindo assim a compatibilidade do nosso código em diferentes tipos de browsers.

O resultado da transcompilação pode variar de transpiler para transpiler, mas o resultado final deve ser idêntico à funcionalidade original do código em ES2015. Inclusive não é raro o resultado da transcompilação para ES5 resulte em um código muito mais verboso.

Vejamos a seguinte hierarquia de projeto:

```
app
  js
    es6
      a.js
  css
    a.css
  img
    logo.png
index.html
```

Nessa estrutura, temos o arquivo `app/js/es6/a.js`. Sabemos que o `index.html` não pode importar diretamente o arquivo, pois se estivermos executando nosso código no IE 9, por exemplo, não teremos suporte para vários recursos do ES6. A ideia é converter o código escrito em ES6 para ES5, por exemplo, resultando na seguinte estrutura:

```
app
  js
    es6
      a.js <-- será transcompilado por um transpiler
  css
```

```
a.css
img
  logo.png
index.html
```

Resultado da transcompilação:

```
app
  js
    es5
      a.js <-- resultado da transcompilação
    es6
      a.js
  css
    a.css
  img
    logo.png
index.html
```

Veja que é o arquivo `app/js/es5/a.js` que deve ser importado em `index.html`. Além disso, qualquer mudança deve ser empreendida no arquivo `app/js/es6/a.js`. Depois de efetuada a alteração, o arquivo precisa ser transcompilado para que `app/js/es5/a.js` reflita a transcompilação da versão mais nova do código.

O processo de transcompilação normalmente não é feito manualmente, mas por meio de ferramentas que tornam transparentes esse processo para o desenvolvedor, evitando assim erros oriundos do esquecimento da compilação deste ou daquele arquivo que foi atualizado.