01

Importando o Bootstrap

Transcrição

É bem provável que você já tenha ouvido falar do Bootstrap, um conjunto de classes e scripts que permitem aplicar um visual profissional em nossa aplicação com pouquíssimo esforço. Mas será que utilizar o Bootstrap em uma Single Page Application criada com o Vue CLI é tão trivial quando adicionar a tag `link` no `head` de `index.html` ?

A solução mais simples, porém não indicada

A solução mais simples, porém não indicada é adicionar dentro da tag `head` de `alurapic/index.html` a tag link que aponte para algum CDN do bootstrap. Por exemplo:

```
<!-- NÃO FAÇA ISSO! É APENAS UM EXEMPLO. FAREMOS DE OUTRA FORMA!!! -->

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>alurapic</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.r
  </head>
  <body>
    <div id="app"></div>
    <script src="/dist/build.js"></script>
  </body>
</html>
```

Simples não? É o que a maioria de tutoriais da internet pedirá que você faça em seu projeto, mas essa abordagem deixa a desejar. O que acontecerá se o CDN estiver fora do ar? Com certeza sua aplicação ficará sem o Bootstrap. Outro ponto é que o arquivo não poderá ser incluído em nosso `bundle.js` que passa por uma série de otimizações.

Vamos abdicar dessa solução e usar uma mais profissional, já que o Vue CLI utiliza Webpack. Vejam, digo mais profissional, não quer dizer a mais fácil.

A solução profissional

Primeiro, precisamos buscar o Bootstrap. Se passou pela sua cabeça ir lá no site do Bootstrap e baixar seus arquivos desista. A prática de baixar dependências front-end manualmente já não mais existe. Podemos usar o `npm` para baixar o bootstrap para nós. Isso mesmo! Se você não esta acostumado com essa prática, bem-vindo ao time.

No terminal, dentro da pasta `alurapic` vamos executar o comando:

```
npm install bootstrap@3.3.7 --save
```

Depois de alguns instantes, será criada a pasta `node_modules/bootstrap` com todos os seus arquivos. Agora, vamos alterar `alurapic/src/main.js` importando o bootstrap como se fosse um módulo:

```
// alurapic/src/main.js

import Vue from 'vue'
import App from './App.vue'
import VueResource from 'vue-resource';
import VueRouter from 'vue-router';
import { routes } from './routes';
import './directives/Transform';
import VeeValidate from 'vee-validate';
import msg from './pt_BR';

import 'bootstrap/dist/css/bootstrap.css';

// código posterior omitido
```

Quando usamos a instrução `import`, nossa aplicação será inteligente para procurar a pasta do bootstap dentro de `node_modules`. Mas infelizmente isso não vai funcionar porque o arquivo `bootstrap.css` não é um módulo. E agora? É ai que entra a mágica do Webpack.

Webpack e loaders

O Webpack é un bundle que com auxílio de outros módulos pode gerar transformações em nossos arquivos para no final gerar um `bundle` otimizado da nossa aplicação. Vimos isso neste capítulo, inclusive aprendemos a criar o bundle para distribuição.

Webpack não fez "curso mãe Diná" para saber como deve carregar e tornar disponível em nosso `bundle` qualquer tipo de arquivo. Ele utiliza `loaders` para essa tarefa. Existem vários `loaders` no mercado, cada um com uma tarefa específica.

No caso, para carregarmos um arquivo `css` que esteja dentro `node_modules` precisamos pedir ajuda a dois `loaders`. O primeiro é o **style-loader** e o segundo o **css-loader**. O que esses loaders farão?

Os loaders que baixaremos farão com que o CSS do bootstrap seja importável através da instrução `import` adicionando-o no final no `bundle` da aplicação.

Pode parecer um exagero, mas essa abordagem nos permite a utilizarmos qualquer dependência de front-end baixada dentro da pasta `node_modules`.

Além do que vimos, essas dependências passarão por uma série de otimizações e transformações que o Webpack configurado pelo Vue CLI esta configurado para fazer. Precisamos baixas os loaders.

No terminal, dentro da pasta `alurapic` vamos baixar os loaders:

Meu aluno, minha aluna, não queiram dar uma de espertos carregando a versão mais novas desses módulos. Na altura do campeonato vocês já devem ter aprendido na pele que a última coisa que se faz em um projeto é atualizar suas dependências. As versões a seguir são amplamente testadas com o Vue CLI. A boa notícia é que são as versões mais atualizadas na data de lançamento deste curso!:

```
npm install css-loader@0.25.0 style-loader@0.13.1 --save-dev
```

Pronto! Baixamos nossos loaders. Agora precisamos configurá-los.