

O módulo boot.js

Transcrição

Conseguimos remover inúmeros scripts do `index.html` :

```
<div id="negociacoesView"></div>

<script src="node_modules/systemjs/dist/system.js"></script>
<script>
  System.defaultJSExtensions = true;
  System.import('js/app/boot').catch(function(err){
    console.error(err);
  });
</script>
```

Se você analisar a `lib` , perceberá que ela não segue o sistema de módulos do ES6. Resolveremos isto mais adiante. Vamos adicionar a variável `negociacaoController` , no entanto, ela está em um escopo privado. Não temos mais o `negociacaoController` no escopo global... Para resolver a questão, não associaremos mais os eventos na View.

Importamos a lib do `system.js` e fizemos a configuração mínima, especificando que omitiremos as extensões dos arquivos e qual será o primeiro módulo a ser carregado. Agora, começaremos a escrever o arquivo `boot.js` :

```
import {NegociacaoController} from './controllers/NegociacaoController';
import {} from './polyfill/fetch';

let negociacaoController = new NegociacaoController();
```

Primeiramente, importamos o `NegociacaoController` e carregamos o polyfill, para que ele seja executado. No entanto, o `negociacaoController` está em um escopo privado. No arquivo `index.html` , por exemplo, chamamos `negociacaoController` e depois o método `apaga()` :

```
<div class="text-center">
  <button onclick="negociacaoController.apaga()" class="btn btn-primary text-center" type="button">
    Apagar
  </button>
</div>
```

Como não temos mais `negociacaoController` no escopo global, deixaremos de associar o evento de onde ele era utilizado diretamente na View. No caso, o trecho referente ao botão "Apagar" ficará assim:

```
<div class="text-center">
  <button class="btn btn-primary text-center" type="button">
    Apagar
  </button>
</div>
```

Em seguida, vamos inserir as classes do `index` no `boot.js`. Usaremos o `querySelector()`:

```
import {NegociacaoController} from './controllers/NegociacaoController';
import {} from './polyfill/fetch';

let negociacaoController = new NegociacaoController();

document.querySelector('.form').onsubmit = negociacaoController.adiciona.bind(negociacaoController);
document.querySelector('[type=button]').onclick = negociacaoController.apaga.bind(negociacaoController);
```

O porquê do uso do `bind` já foi visto anteriormente. Se não o incluíssemos, o `this` do `adiciona` não seria mais o `negociacaoController`. Escolhemos uma solução tradicional.