▶ 01
API externa

Transcrição

Começando deste ponto? Você pode fazer o [download (https://s3.amazonaws.com/caelum-online-public/typescript/09-alurabank.zip)](https://s3.amazonaws.com/caelum-online-public/typescript/09-alurabank.zip) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo.

É muito comum aplicações acessarem API's de terceiros, com nossa aplicação não será diferente. Temos um servidor escrito em Node.js que disponibilizará a API que consumiremos e ele pode ser baixado [aqui (https://s3.amazonaws.com/caelum-online-public/typescript/api.zip)](https://s3.amazonaws.com/caelum-online-public/typescript/api.zip). Após descompactar o arquivo, teremos a pasta `api` . Como levantar o servidor?

Basta abrirmos outro terminal (um é para o nosso lite-server outro para o servidor da API) e dentro da pasta `api` executarmos o comando:

```
npm start
```

Através do endereço `http://localhost:8080/dados` temos acesso a um JSON com a seguinte estrutura:

```
[{"montante":200.5,"vezes":2},{"montante":100.2,"vezes":5},{"montante":50.5,"vezes":1},{"montan
```

Nele, temos os dados legados de negociações de outro sistema. Nele, não há o campo `data` e os campos `quantidade` e `valor` se chamam `vezes` e `montante` respectivamente. Nosso cliente disse que podemos usar a data da exportação como data para termos os dados suficientes de uma negociação. É o velho papel de integração entre sistemas. Em suma, nosso objetivo será consumir essa API, criar uma lista adequada de negociações e adicioná-las na tabela que é exibida para o usuário.

Primeiro, vamos adicionar o botão `importar` em `app/index/.html` :

```html
<!- app/index.html -->
<!-- código anterior omitido -->
        <button class="btn btn-primary" type="submit">Incluir</button>
        <button id="botao-importa" class="btn btn-primary" type="button">Importar</button>
    </form>
<!-- código posterior omitido -->
```

Agora, em `NegociacaoController` , vamos criar o método `importarDados` que apenas exibe um alerta quando chamado:

```ts
// app/ts/controllers/NegociacaoController.ts

    importarDados() {

        alert('oi');
```

```
    }
```

Agora, em `app/ts/app.ts` vamos associar o clique do botão com a chamada do novo método:

```typescript
import { NegociacaoController } from './controllers/NegociacaoController';

const controller = new NegociacaoController();
$('.form').submit(controller.adiciona.bind(controller));
$('#botao-importa').click(controller.importarDados.bind(controller));
```

Perfeito, acessando nossa aplicação, ao clicarmos no novo botão, o alerta é exibido. Agora que temos a estrutura pronta, podemos dar início à importação dos dados da API.