01

Serviços disponíveis

Transcrição

> Você pode fazer o download (https://github.com/alura-cursos/javascript-avancado-ii/archive/aula3.zip) completo do projeto até aqui e continuar seus estudos.

Nós aplicamos diversos recursos da linguagem JavaScript, padrões de projetos e outros... Mas para a aplicação ficar completa, além de incluir negociações, queremos poder importar negociações de serviços na web. Alguns bancos oferecem a possibilidade de obter as negociações da semana atual e da anterior. Nós já disponibilizamos no projeto uma pasta chamada `server`. No primeiro exercício obrigatório (https://cursos.alura.com.br/course/javascript-es6-orientacao-a-objetos-parte-2/task/23371) do curso, você encontrará toda a infraestrutura que você precisa para subir o servidor.

Com a infraestrutura instalada, entraremos na pasta `Desktop`. Para isto, você deve ter um conhecimento básico sobre o uso do Terminal para entrar nesta pasta.

> No curso, usaremos o Terminal do Mac. Mas se você é usuário de Windows e não domina o prompt de comando, nossa recomendação é que faça o curso Windows: Introdução ao Prompt (https://cursos.alura.com.br/course/prompt). Neste ponto do curso, assumiremos que você já sabe interagir com seu Terminal.

Para levantar o servidor, abriremos nosso Terminal e depois, entraremos na pasta `aluraframe/server`. É a primeira vez que interagiremos com essa pasta, mas basicamente, você só precisa rodar o comando no seu terminal:

```
> npm start
```

Caso esse comando de erro use esse:

```
> node server.js
```

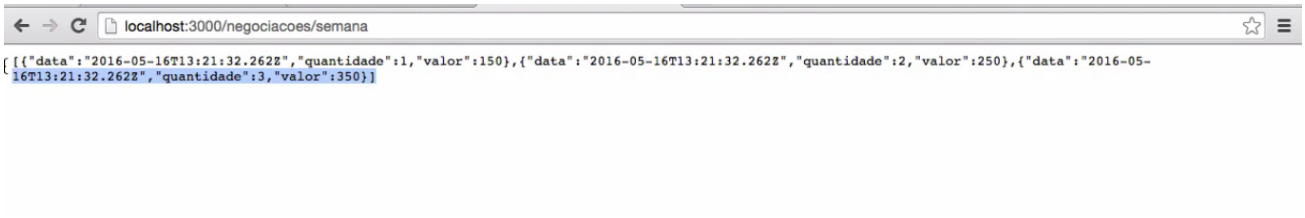
Isto fará com que um servidor rode e seja acessível por meio do endereço `http://localhost:3000`. Acesse esse endereço e automaticamente a página `index.html` será carregada. Se preferir, pode digitar `http://localhost:3000/index.html`.

A única mudança até agora é não acessar mais `index.html` do sistema de arquivos da plataforma, mas por intermédio do servidor. Certifique-se que o servidor esteja funcionando antes de continuar. O nosso servidor além de servir o `index.html`, possui também alguns serviços.

Geralmente, quem publica serviços na web disponibiliza uma URL de acesso. O nosso servidor publicará três:

```
negociacoes/semana
negociacoes/anterior
negociacoes/retrasada
```

Ao acessarmos o endereço `http://localhost:3000/negociacoes/semana` , sairemos da página `index.html` e, na tela, será exibida uma estrutura de dados no formato JSON (JavaScript Object Notation):



No entanto, não queremos exibir o arquivo JSON na tela e buscar os dados por meio de URL. Nosso objetivo é que ao clicarmos no botão `Importar Negociações` , seja feita a busca dos dados usando JavaScript e depois, os dados sejam inseridos na tabela de negociações. Desta forma, a tabela será automaticamente atualizada.

Nós temos todo mecanismo de *Data binding*, então, basta incluirmos os dados na tabela. A seguir, nós iremos relembrar como acessamos endereços na Web por meio do JavaSript, usando AJAX. Faremos uma revisão sobre o assunto e assim, entender o problema que desejamos resolver e qual recurso avançado pode nos ajudar em programação assíncrona .

> O foco do curso é sobre como consumimos os dados publicados pelos servidores usando o JavaScript. Se você deseja aprender como servidores são criados e como eles disponibilizam esse tipo de dados, nossa sugestão é que você faça o curso de **MEAN** (https://cursos.alura.com.br/course/mean-javascript).