

Desafio

Módulo 4	Fundamentos em Front-End
----------	--------------------------

Objetivos

Exercitar os seguintes conceitos trabalhados no Módulo:

- ✓ Programação em JavaScript.
- ✓ Interação com o DOM.
- ✓ Manipulação de arrays.
- ✓ Requisições HTTP.
- ✓ Estilização com CSS.

Enunciado

Construa, utilizando HTML, CSS e JavaScript, uma página de listagem de registros, com ordenação e filtros configuráveis, utilizando o Back End fornecido junto com o código fonte das aulas.

Atividades

Os alunos deverão desempenhar as seguintes atividades:

1. Crie uma página para sua aplicação e utilize *fetch* para buscar os dados dos dois *endpoints* disponibilizados: `/employees` e `/roles` (não se esqueça de executar o comando `npm install` antes de iniciar o Back End pela primeira vez).
2. Exiba a lista `employees` em uma tabela, contendo as seguintes colunas:

- a. ID (campo id);
 - b. Name (campo name);
 - c. Role (Campo name da *role* correspondente ao campo *role_id*);
 - d. Salary (campo salary).
3. Inclua na aplicação um elemento *select* que permita escolher um critério de ordenação. Os dados devem ser inicialmente ordenados alfabeticamente pelo Nome. Sempre que o valor do *select* mudar, eles devem ser reordenados usando JavaScript (não busque dados do Back End). As opções de ordenação são:
 - a. Nome ascendente (valor padrão).
 - b. Nome descendente.
 - c. Salário ascendente.
 - d. Salário descendente.
4. Inclua também na aplicação a opção de filtrar os registros pela coluna Role. Para cada objeto obtido em */roles*, inclua um *checkbox* correspondente. Ao marcar um ou mais checkboxes, a tabela deve exibir apenas *employees* com qualquer uma das roles selecionadas. Por exemplo, se estão marcados Data Analyst e Project Manager, exiba apenas *employees* com role Data Analyst **OU** Project Manager. Se não houver nenhuma *role* marcada, exiba todos os registros. A filtragem também deve ser feita via JavaScript (sem buscar dados do Back End).
5. Exiba na interface da aplicação a quantidade de registros exibidos no momento, para facilitar a contagem.
6. Estilize a interface como desejado, mas cumprindo duas exigências:
 - a. As opções de ordenação e filtros devem ficar ao lado da tabela de dados, à esquerda dela.

- b. A coluna Salary deve ter seu conteúdo alinhado à direita.

Figura 1 - Exemplo de interface.

Options		Employees (23)			
Sort by salary (ascending) ▼		ID	Name	Role	Salary
Filter by roles		25	Mayor Foucar	Senior Web Developer	2200
<input type="checkbox"/> Database Administrator		177	Kikelia Le Lievre	Senior Web Developer	2400
<input type="checkbox"/> Senior QA Analyst		141	Evangeline Damsell	Senior Web Developer	3000
<input type="checkbox"/> Junior QA Analyst		174	Karney Hatry	Senior Web Developer	3200
<input type="checkbox"/> Senior Web Designer		192	Lucian Upward	Senior Web Developer	3600
<input type="checkbox"/> Junior Web Designer		8	Caro Essberger	Senior Web Developer	3800
<input type="checkbox"/> Junior Web Developer		88	Latia Dowthwaite	Senior Web Developer	3800
<input checked="" type="checkbox"/> Senior Web Developer		81	Virginia Limbert	Senior Web Developer	4200
<input type="checkbox"/> Data Analyst		107	Nanni Quogan	Senior Web Developer	4800
<input type="checkbox"/> Junior Devops Engineer		70	Maurizio Larnar	Senior Web Developer	5200
<input type="checkbox"/> Senior Devops Engineer		180	Yasmin Burnyate	Senior Web Developer	5400
<input type="checkbox"/> Network Administrator		116	Cynde Bumphrey	Senior Web Developer	6600
<input type="checkbox"/> Project Manager		46	Fredrika Vasilkov	Senior Web Developer	6800
		60	Elinor Josskoviz	Senior Web Developer	6800
		186	Matthieu Ruane	Senior Web Developer	7200
		66	Roth Glancy	Senior Web Developer	7600
		108	Shirlene Masey	Senior Web Developer	7600
		75	Fran Synder	Senior Web Developer	7800
		128	Gail Giotto	Senior Web Developer	7800
		96	Charisse Witling	Senior Web Developer	8400
		196	Fayth Rosentholer	Senior Web Developer	8600

Dicas

- Existem várias maneiras de resolver o problema. As dicas a seguir podem ser relevantes ou não, dependendo da sua abordagem.
- Lembre-se que o elemento *option* possui a propriedade *value*, e que a propriedade *value* do elemento *select* corresponde ao *value* do *option* selecionado.
- Checkboxes também possuem a propriedade *value*, assim como elementos *option*.
- O evento *input* não é disparado em *checkbox* ou *select*, mas você pode usar o evento *change* para reagir a mudanças nos valores.
- É possível determinar se um checkbox está marcado pela propriedade *checked*. Além disso, existe uma pseudoclassee *:checked* que pode ser usada para montar

um seletor CSS que captura checkboxes marcados. Você pode usá-la em conjunto com `document.querySelectorAll`.

- Você pode gerar a tabela via strings e a propriedade `innerHTML`, ou criando elementos via `document.createElement`. No segundo caso, usar o `innerHTML` para limpar os dados antes de gerá-los novamente pode te ajudar a deixar o código mais simples.
- Lembre-se que *arrays* possuem a função *sort*, que recebe uma função de comparação como parâmetro. Essa função deve retornar um número negativo, zero ou positivo para indicar, respectivamente, se o primeiro parâmetro é menor, igual ou maior que o segundo. As funções *filter*, *includes* e *indexOf* também podem ser de interesse.
- O *flexbox* é provavelmente o mecanismo mais fácil de montar seu layout, mas existem outras formas.
- As pseudoclasses `:first-child` ou `:last-child` podem te ajudar a estilizar a tabela sem precisar recorrer a IDs ou classes.
- Se você quiser adicionar bordas em elementos *tr*, *thead* ou *tbody*, precisará aplicar na tabela a propriedade `border-collapse: collapse`;