

Vaga: Estágio – TI: Web Analytics

Candidato: Caio Teruo Fuzita Rabelo da Silva

1. Módulos

1.1. File System

```
1 fs = require('fs');
```

Necessário para abrir arquivos, modificar e salvar.

2. Funções

2.1. Abrir arquivo .json

```
40 function readJson(){
41   const data = fs.readFileSync('./broken-database.json','utf8'); // abre o arquivo JSON
42   return JSON.parse(data); // transforma o JSON em objeto javascript
43 }
```

Cria uma constante com os dados do arquivo 'broken-database.json', e para que seja possível manipular os dados, é utilizado o método .parse, que transforma o arquivo .json em objeto Javascript.

2.2. Correção de nomes

```
46 function fixName(str){
47   return str.replace(/æ/g, 'a').replace(/ç/g, 'c').replace(/ø/g, 'o').replace(/ß/g, 'b'); //https://www.
48   devmedia.com.br/javascript-replace-substituindo-valores-em-uma-string/39176
}
```

Utiliza o método .replace com o parâmetro global (/g) para substituir os caracteres corrompidos.

2.3. Correção das quantidades

```
51 function fixQuantity(quantity){
52   return quantity || 0; // retorna o valor de quantity, e se não houve um valor, retorna 0;
53 }
```

Retorna o próprio valor do 'quantity', se não houve um valor, ele retornará 0.

2.4. Correção dos preços

```
56 function fixPrice(price){
57   return Number(price); /
58 } //https://developer.mozil
```

Retorna a variável como tipo 'number'.

2.5. Salvar o arquivo

```
61 function writeJson(file){
62     var dados = JSON.stringify(file); // transforma o objeto javascript em .json
63     fs.writeFileSync('./saida.json', dados);
64 }
```

Primeiramente transforma o objeto Javascript de volta para o tipo .json, após isso, salva o arquivo com o nome 'saida.json'.

2.6. Abrir o arquivo corrigido

```
67 function readFixedJson(){
68     const dataFixed = fs.readFileSync('./saida.json', 'utf8');
69     return JSON.parse(dataFixed);
70 }
```

Executa a mesma função de abrir o arquivo tipo .json (cria uma constante com o arquivo e retorna como objeto .js) com o arquivo já corrigido.

2.7. Verificar e listar categorias únicas

```
73 function verifyCategory(){
74
75     var categorias = []; // criação de um array vazio
76     for (var i in fixedJson){
77         categorias.push(fixedJson[i].category); // atribui todas as categorias existentes no
78         // array 'categorias'
79     }
80
81     var categoriasUnicas = []; // criação de um array vazio
82
83     for (var i = 0; i < categorias.length; i++){
84         if (categorias[i] != categorias[i+1]){
85             categoriasUnicas.push(categorias[i]); // varredura por todos os elementos da array
86             // 'categorias', todos os valores únicos vão para o 'categoriasUnicas'
87         }
88     }
89 }
```

Primeiramente a função cria um array vazio, e com o comando de laço atribui todas as categorias existentes no arquivo .json para o array criado, logo após, cria outra array e com outro comando de laço verifica as categorias e retorna ao array apenas as categorias que não se repetem.

2.8. Calcular estoque

```
91 function calculadora(products){
92     var soma = 0; // iniciador da soma
93     for (var i in products){
94         soma += products[i]['quantity'] * products[i]['price']; // multiplica a quantidade
95         // pelo preço
96     }
97     return soma
98 }
```

Inicia a variável soma e com um comando de laço multiplica a quantidade de produtos do argumento pelo preço dele, no fim retorna o valor da soma.

3. Referências

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/sort

<https://nodejs.org/api/fs.html#fsreadfilepath-options-callback>

<https://www.devmedia.com.br/javascript-replace-substituindo-valores-em-uma-string/39176>

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/parseInt