



FULL DIGITAL PERFORMANCE

Projeto Classificatório

Processo seletivo - Analista em Tecnologia

Documentação do projeto

- Funções

1. lerJSON:

```
function lerJSON(filePath){
  const fs = require('fs');
  try{
    const data = fs.readFileSync(filePath);
    var produtos = JSON.parse(data);
  }catch(e){
    console.log(e);
  }
  return produtos;
}
```

Responsável por fazer a leitura do arquivo broken-database.json e retornar seu conteúdo como um objeto JavaScript.

2. concertaNomeJSON:

```
function concertaNomeJSON(){
  var produtos = lerJSON('./broken-database.json');
  produtos.forEach(function(produto){
    var newString = produto.name.replace(/æ/g, "a");
    newString = newString.replace(/ç/g, "c");
    newString = newString.replace(/ø/g, "o");
    newString = newString.replace(/ß/g, "b");
    produto.name = newString;
  });
  return produtos;
}
```

Substitui os caracteres falhos (æ, ç, ø, ß) do objeto JavaScript pelos caracteres corretos (a, c, o, b), retornando um objeto JavaScript atualizado.

3. concertaPrecoJSON:

```
function concertaPrecoJSON(produtos){
  var aux = '';
  produtos.forEach(function(produto){
    if(typeof produto.price == typeof aux){
      produto.price = Number(produto.price);
    }
  });
  return produtos;
}
```

Tem como objetivo alterar o tipo do valor do atributo “price”, do objeto JavaScript, para o tipo number, caso este esteja no tipo string. Retornando um objeto JavaScript atualizado.

4. concertaQuantidadeJSON:

```
function concertaQuantidadeJSON (produtos){
  produtos.forEach(function(produto){
    if(produto.quantity == undefined){
      produto.quantity = 0;
    }
  });
  return produtos;
}
```

Responsável por inserir um atributo “quantity” com o valor “0” em um elemento do objeto JavaScript, caso este não possua tal atributo. Também retornando o objeto JavaScript atualizado.

5. exportaJSON:

```
function exportaJSON(){
  console.log('JSON exportado:');
  const fs = require('fs');
  var produtosFinal = concertaQuantidadeJSON(concertaPrecoJSON(concertaNomeJSON()));

  try{
    const saida = JSON.stringify(produtosFinal);
    fs.writeFileSync('./saida.json', saida);
  }catch(e){
    console.log(e);
  }
  console.log(produtosFinal);
}
```

Obtém o objeto JavaScript completamente corrigido pelas funções anteriores, o converte novamente para JSON e o exporta para um arquivo chamado saida.json. Por fim, imprime no console o objeto JavaScript.

6. imprimeNomeOrdemAlfabetica:

```
function imprimeNomeOrdemAlfabetica(){
  console.log('\nImprimindo nome dos produtos na ordem requisitada:');
  var produtosFinal = concertaQuantidadeJSON(concertaPrecoJSON(concertaNomeJSON()));
  var nomeProdutos = [];
  var aux = '';
  produtosFinal.forEach(function(produto){
    aux = produto.category + ': id (' + produto.id + ') - ' + produto.name;
    nomeProdutos.push(aux);
  });
  nomeProdutos.sort(function(a, b) {
    if(a < b){
      return -1;
    }
    else{
      return true;
    }
  });
  console.log(nomeProdutos);
}
```

Essa função de validação imprime o nome dos produtos por categoria em ordem alfabética e por id em ordem crescente. Assim, garante que o nome dos mesmos foi corrigido pela função concertaNomeJSON.

7. validaEstoque:

```

function validaEstoque(){
  console.log('\nValidando estoque...');
  var produtosFinal = concertaQuantidadeJSON(concertaPrecoJSON(concertaNomeJSON()));
  let w = 0, x = 0, y = 0, z = 0;
  produtosFinal.forEach(function(produto){
    if(produto.category == 'Painéis'){
      w += produto.price * produto.quantity;
    }
    if(produto.category == 'Eletrodomésticos'){
      x += produto.price * produto.quantity;
    }
    if(produto.category == 'Eletrônicos'){
      y += produto.price * produto.quantity;
    }
    if(produto.category == 'Acessórios'){
      z += produto.price * produto.quantity;
    }
  });
  console.log('Valor do estoque das painéis:', w);
  console.log('Valor do estoque dos eletrodomésticos:', x);
  console.log('Valor do estoque dos eletrônicos:', y);
  console.log('Valor do estoque dos acessórios:', z);
}

```

Essa função de validação imprime o valor total do estoque de cada categoria de produto, garantindo com que as funções `concertaPrecoJSON` e `concertaQuantidadeJSON` tenham cumprido com seus objetivos.

- Por que JavaScript?

JavaScript foi a linguagem requisitada para o projeto. Sendo a linguagem mais adequada para tratar da notação JSON.

- Executando a aplicação

Para executar a aplicação, basta rodar no terminal `node resolucao.js`. Todas as informações relevantes estarão no terminal após o término da execução.