

Processo Seletivo - TI

Web Analytics

Rocky

Nome: André Luis de Souza Oliveira

Funcionamento do código JavaScript

O código possui 4 funções, que são: `ler_json()`, `corrige_nome()`, `corrige_vendas()` e `export_json()`. Para iniciar, o programa chama a função `ler_json()` a fim de obter o arquivo `broken_database_1.json` e `broken_database_2.json`. Após isso, com um `if()` verificamos se foi possível ler os arquivos e passamos os objetos dos bancos de dados como parâmetros para as outras funções.

`ler_json()`: Esse método tenta fazer um requisição com a função `require()` dos arquivos JSONs, após isso, transforma esse arquivos JSONs em um objeto JavaScript, por fim retorna esses 2 objetos como somente um objeto.

Em seguida é chamada a função `corrige_nome()`, Por meio da função `replace()` encadeada, juntamente do uso de expressões regulares é trocado os caracteres æ e ø por “a” e “o” respectivamente, presentes no nome e marcas dos carros que estão nos objetos a serem corrigidos.

A função `corrige_vendas()`, usando `typeof` é comparado os valores de venda com o tipo `String`, se verdadeiro, transformamos os valores das vendas em valores do tipo `Number`.

Por fim temos a função `export_json()`, que cria uma requisição ‘fs’ para guardar o arquivo no disco, logo após chama a função `writeFile()`, onde especifica o nome do novo arquivo e o que será armazenado nele, que no caso são os objetos modificados transformados para JSON.

Tratamento de exceções

Dois possíveis erros de ocorrerem durante a execução do código estão relacionados a leitura dos arquivos e a escrita das modificações em um arquivo. Ao tentar ler o arquivo no `ler_json()`, seria possível não achar os arquivos “`broken_database_1.json`” e “`broken_database_2.json`”. Para que o programa não pare de funcionar, o console deve informar que houve um erro ao tentar acessar o arquivo e na linha de baixo printar o erro que foi gerado. Foi possível realizar esse tratamento por meio de um `try/catch`, em que, ao entrar no `catch` a função `ler_json()` retorna `null`, então, para se continuar a execução do código, verifica se a variável `databases` não é nula.

O outro possível erro é relacionado ao salvamento do novo arquivo json, na função `export_json()`. Para esse tratamento, há uma função ao escopo do arquivo `.writeFile(..., function(err))` com um `throw(err)` dentro dessa função.

Código SQL

O código SQL foi feito no Postgresql. Após a importação dos JSONs para o SQL, feito as alterações dos nomes das colunas das tabelas, a criação de uma nova tabela através da junção das tabelas `venda_carro` e `marca`. O código SQL para a criação da tabela `venda_carro_marca` está abaixo:

```
CREATE TABLE venda_carro_marca AS  
SELECT * FROM marca NATURAL JOIN venda_carro
```