Bootcamp Desenvolvedor(a) Node.js

Desafio do módulo

Módulo 1	Desenvolvimento Back-End com JavaScript

Objetivos

Exercitar os conceitos trabalhados no módulo para criação de uma API, criando endpoints utilizando Node.js e Express.

Enunciado

Desenvolver uma API chamada “delivery-api” para controlar pedidos de um delivery de comida.

Atividades

O desafio final consiste em desenvolver uma API chamada “delivery-api” para controlar pedidos de clientes de um delivery. Você deverá desenvolver endpoints para criação, atualização, exclusão e consulta de pedidos. Os pedidos deverão ser salvos em um arquivo json chamado “pedidos.json”. Este arquivo será previamente fornecido e seus endpoints devem atuar considerando os registros já existentes.

Um pedido deve possuir os campos abaixo:

- Id (int): identificador único do pedido. Deve ser gerado automaticamente pela API e deve ser garantido que este não se repita.
- Cliente (string): nome do aluno. Exemplo: “Guilherme Assis”.
- Produto (string): nome do produto. Exemplo: “Pizza de Calabresa”.
- Valor (float): valor do produto. Exemplo: 10.

- Entregue (boolean): indica se o pedido já foi entregue (true = foi entregue, false = ainda não foi entregue). Exemplo: true.
- Timestamp (string): horário de criação do pedido. Exemplo: 2020-05-19T18:21:24.964Z. Dica: utilizar o “new Date()” do JavaScript.

O arquivo pedidos.json será previamente fornecido com alguns registros inseridos. Seus endpoints devem trabalhar considerando a existência deles, não devendo ser criado um arquivo limpo para utilização. A estrutura do arquivo é a seguinte:

```
{
  "nextId": 49,
  "grades": [
    {
      "id": 1,
      "student": "Loiane Groner",
      "subject": "01 - JavaScript",
      "type": "Fórum",
      "value": 15,
      "timestamp": "2020-05-19T18:21:24.958Z"
    }
    //... demais registros
  ]
}
```

A propriedade “nextId” deve armazenar sempre o próximo id que será utilizado na criação de um novo pedido. A propriedade “pedidos” possui um array com vários pedidos, cada um sendo representado por um objeto com os campos descritos anteriormente. Para facilitar a criação da API, cada pedido possui somente um produto vinculado.

Você deverá desenvolver os endpoints descritos abaixo:

1. Crie um endpoint para criar um pedido. Este endpoint deverá receber como parâmetros os campos cliente, produto e valor conforme descritos acima. Este pedido deverá ser salvo no arquivo json ‘pedidos.json’ e deverá ter um id único associado. No campo “timestamp”, deverão ser salvos a data e a hora do momento da inserção. O campo “entregue” deverá ser criado inicialmente como “false”, pois ele poderá ser atualizado posteriormente através de outro endpoint. O endpoint deverá retornar o objeto do pedido que foi criado.

   A API deverá garantir o incremento automático deste identificador, de forma que ele não se repita entre os registros. Dentro do arquivo pedidos.json que foi fornecido

para utilização no desafio, o campo nextId já está com um valor definido. Após a inserção, é preciso que esse nextId seja incrementado e salvo no próprio arquivo, de forma que na ele possa ser utilizado próxima inserção.

2. Crie um endpoint para atualizar um pedido. Este endpoint deverá receber como parâmetros o id do pedido a ser alterado e os campos “cliente”, “produto”, “valor” e “entregue”. O endpoint deverá validar se o produto informado existe. Caso não exista, ele deverá retornar um erro; caso exista, o endpoint deverá atualizar as informações recebidas por parâmetros no registro e realizar sua atualização com os novos dados alterados no arquivo pedidos.json.

3. Crie um endpoint para atualizar o status de entrega do pedido, alterando o campo “entregue” de acordo com o parâmetro informado. Este endpoint deverá receber como parâmetros o id do pedido a ser alterado e o novo valor para o campo “entregue”, sendo os valores possíveis true ou false. Este endpoint deverá atualizar somente o valor do campo “entregue” do registro de ID informado, alterando-o no arquivo pedidos.json.

4. Crie um endpoint para excluir um pedido. Este endpoint deverá receber como parâmetro o id do pedido e realizar sua exclusão no arquivo pedidos.json.

5. Crie um endpoint para consultar um pedido em específico. Este endpoint deverá receber como parâmetro o id do pedido e retornar suas informações.

6. Crie um endpoint para consultar o valor total de pedidos já realizados por um mesmo cliente. O endpoint deverá receber como parâmetro o cliente, realizar a soma dos valores de todos os seus pedidos e retornar essa informação. O endpoint deve considerar somente os pedidos já entregues.

7. Crie um endpoint para consultar o valor total de pedidos já realizados para um determinado produto. O endpoint deverá receber como parâmetro o produto, realizar a soma dos valores de todos os pedidos deste produto específico e retornar essa informação. O endpoint deve considerar somente os pedidos já entregues.

8. Crie um endpoint para retornar os produtos mais vendidos e a quantidade de vezes em que estes foram pedidos. O endpoint não deve receber parâmetros. O endpoint deve calcular os produtos que mais possuem pedidos e retorná-los em ordem

decrescente, seguidos pela sua quantidade. exemplo: [“Pizza A - 30”, “Pizza B – 27”, “Pizza C – 25”, “Pizza D – 23”, “Pizza E – 21”, “Pizza F – 19”, “Pizza G – 17”]. O endpoint deve considerar somente os pedidos já entregues.

Respostas Finais

Os alunos deverão desenvolver a prática e, depois, responder às questões objetivas.

IMPORTANTE: Responder as questões na ordem e com o arquivo JSON do jeito que foi fornecido, sem efetuar alterações prévias, de forma a evitar dados inconsistentes.