

Avaliação - Programação



Olá, candidato!

Queremos conhecer um pouco mais sobre sua forma de desenvolver aplicações e resolver problemas, para isso preparamos um pequeno desafio.

O desafio 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.

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.

O seu desafio é construir uma aplicação que gerencia dados de pedidos. Essa aplicação será um CRUD com API (backend) para gerenciamento de pedidos.

Todos os pedidos devem possuir:

- identificador único do pedido (int). Deve ser gerado automaticamente pela API e deve ser garantido que este não se repita.
- nome do cliente (string).
- nome do produto (string).
- valor do produto (float).
- indica se o pedido já foi entregue (boolean)
- horário de criação do pedido (timestamp).

*** A linguagem de programação deve ser Python3. ***

O que esperamos?

- Uma API simples, que atenda aos requisitos descritos;
- Containerização da aplicação (Docker);
 - Imagem docker para a aplicação;
 - Executar a aplicação em um container.
- Criar um backlog de tarefas a ser realizada, para nível de organização;
 - Backlog deve estar ordenado por ordem de execução de tarefas;
 - Será um diferencial a especificação dos critérios usados para as tarefas.
- README.md contendo informações básicas do projeto e como executá-lo.

Avaliação - Programação



Critérios de Aceitação

Queremos avaliar sua capacidade de desenvolver uma API.

- Deve ser criada uma API que cadastre, liste e edite os pedidos. A API deve ser real e escrita por você. Ferramentas que criam APIs automaticamente (Loopback, json-server, etc) não são aceitas;
- Todos os requisitos acima devem ser cumpridos, seguindo o padrão REST, utilizar o JSON com tipo de dado;
- O teste deve ser enviado para seu repositório no Git e você deve responder este e-mail com o link do projeto para podermos baixá-lo e utilizá-lo segundo a proposta da questão;
- Deixe instruções para que possamos rodar as aplicações.

Você terá **7 dias para realizar**, sendo a deadline de entrega às 23h59 da próxima segunda-feira (14/02). Mas se finalizar antes, pode nos mandar.

Caso você tenha qualquer dúvida, pode mandar por aqui também que iremos encaminhar para o nosso time de desenvolvedores.

Preparado?

Aguardamos seu teste. Boa sorte!

Orientações

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. **Endpoint para criar um pedido.** Deverá receber como parâmetros os campos cliente, produto e valor conforme descritos acima. Este pedido deverá ser salvo no arquivo json e deverá ter um id único associado. A API deverá garantir o incremento automático deste identificador, de forma que ele não se repita entre os registros. No campo "timestamp", deverão ser salvos a data e a hora do momento da inserção. O campo "entregue" deverá ser criado inicialmente como "false", esse poderá ser atualizado posteriormente através de outro endpoint. O endpoint deverá retornar o objeto do pedido que foi criado. 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 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 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.

Avaliação - Programação



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. O endpoint deve considerar somente os pedidos já entregues.