

Olá, candidato, bem-vindo!

Queremos conhecer um pouco mais sobre a sua forma de desenvolver aplicações e resolver problemas. Por isso, preparamos um pequeno desafio, conforme descrito a seguir:

DESAFIO 1

Contexto:

A resposta de um determinado endpoint de API de ERP para uma loja de um restaurante está no arquivo [ERP.json](#).

Este exemplo corresponde a um determinado pedido (guestCheckId) com um único item (guestCheckLineItemId), referente a um único item de menu (menuItem).

Link: [ERP.json](#)

1. Descreva o esquema JSON correspondente ao exemplo acima.
2. Contexto

No exemplo fornecido, o objeto detailLines contém um menuItem. Ele também pode conter instâncias de:

- discount
- serviceCharge
- tenderMedia
- errorCode

Transcreva o JSON para tabelas SQL. A implementação deve fazer sentido para operações de restaurante.

3. Descreva a abordagem escolhida em detalhes. Justifique a escolha.

Recomendações:

- Faça um esboço antes de começar.
- Considere que esta tarefa abrange toda a cadeia de restaurantes.
- Descreva seu processo de pensamento, preocupações e as consequências detalhadamente.
- Certifique-se de incluir pelo menos uma linha por ativo (asset).
- Esperamos ver um código que você estaria confortável em colocar em produção.
- Não hesite em pedir esclarecimentos ou, se preferir, faça uma suposição e siga com ela.

DESAFIO 2

Contexto:

Nossa equipe de inteligência de negócios precisa analisar a receita de todas as lojas de uma rede de restaurantes.

Essas informações podem ser obtidas por meio de 5 endpoints de API.

Method	API/Endpoint	Payload
POST	/bi/getFiscalInvoice	<ul style="list-style-type: none">• busDt: string(date)• storeId: string
POST	/res/getGuestChecks	<ul style="list-style-type: none">• busDt: string(date)• storeId: string
POST	/org/getChargeBack	<ul style="list-style-type: none">• busDt: string(date)

		<ul style="list-style-type: none"> ● storeId: string
POST	/trans/getTransactions	<ul style="list-style-type: none"> ● busDt: string(date) ● storeId: string
POST	/inv/getCashManagementDetails	<ul style="list-style-type: none"> ● busDt: string(date) ● storeId: string

storeId corresponde ao identificador da loja.

busDt corresponde à data de operação.

Tarefa: Esta é uma continuação do primeiro desafio. Você deve armazenar os dados das respostas das APIs (JSON) no nosso data lake.

1. Por que armazenar as respostas das APIs?
2. Como você armazenaria os dados? Crie uma estrutura de pastas capaz de armazenar as respostas da API. Ela deve permitir manipulação, verificações, buscas e pesquisas rápidas.
3. Considere que a resposta do endpoint `getGuestChecks` foi alterada, por exemplo, `guestChecks.taxes` foi renomeado para `guestChecks.taxation`. O que isso implicaria?

Recomendações:

- Faça um esboço antes de começar.
- Considere que esta tarefa abrange toda a cadeia de restaurantes.
- Descreva seu processo de pensamento, preocupações e as consequências detalhadamente.
- Certifique-se de incluir pelo menos uma linha por ativo (asset).
- Esperamos ver um código que você estaria confortável em colocar em produção.
- Não hesite em pedir esclarecimentos ou, se preferir, faça uma suposição e siga com ela.

Você tem até 7 (sete) dias para completar esse desafio, a partir da data de recebimento. Cada arquivo produzido na solução deve estar em um repositório Git, tais quais o código do projeto, README.md, e o documento Kanban. Envie um email para talentos.ti@cocobambu.com com o assunto **Desafio Engenharia Dados Coco Bambu 2024**. O corpo do email deve conter o link para o repositório Git. Com esse repositório, devemos ser capazes de acessar o código, executar o projeto e possíveis testes.

Preparado?

Estamos ansiosos para receber sua solução. Boa sorte!