



Programação Java

Consumo de APIs - Uso de Serviços

// Prática

Objetivo

O objetivo desta prática é exercitar como consumir informações de um determinado microserviço.

¡Boa sorte! 😊👍



Exercício 1

O item API é uma das APIs mais importantes do marketplace do Mercado Livre. Cada publicação feita por um vendedor MeLi se obtém esta API usando endpoints da seguinte forma:

curl https://api.mercadolibre.com/items/<item_id>

Sabemos que cada publicação tem um preço em uma moeda específica, existe uma API que resolve a conversão entre moedas ([link](#) para a documentação pública).

Portanto, precisamos implementar uma API que receba um ID de item e vá procurar o preço original da moeda para esse ID de item, recupere a conversão com a moeda de origem e retorne o preço em dólares para esse item.

Nossa API receberá um GET para um endpoint `/items/prices/[item-id]` e deverá retornar

Temos então que responder as seguintes perguntas (em equipe):



- Como seria o fluxo de execução a partir do momento em que recebemos um request?
- O que aconteceria se recebêssemos 1000 pedidos por minuto? E se recebermos 100.000? Esta API tem uma implementação correta? Que perguntas vocês fariam para pensar em otimizações a serem feitas à API?
- Assumindo que as citações mudam uma vez por dia, que opção podemos tomar para melhorar esta API?



Exercício 2 (Dia 2)

A partir do exercício anterior, tomar como base a implementação na branch `feature/add-currency-conversion-service` da aplicação `bootcamp-demo-java-app` para implementar as otimizações discutidas na aula prática do dia 1.

Parte 1 (em equipes de 6):

- Que parte das informações poderíamos ter pre-cacheada, sem ter perdido a precisão no cálculo?
- Qual seria a janela de exposição para se ter um erro de cálculo no preço em USD? E como podemos ajustá-la?
- Como se implementaria esse cache?
- Que serviços se utilizaria para implementá-lo?
- Indiquem quais classes se modificaria para realizar estas mudanças.

Demo time!!! Ou: hora da demonstração!!! Mostramos uma implementação específica da KVS. Funcionamento geral do SDK e a integração com um serviço específico que consome as informações.

Parte 2 (em equipes de 6):

- Como vocês fariam para evitar ir "ao vivo" para a API de items?
- O que acontece se o preço de um item mudar?
- Como podemos descobrir imediatamente quando o preço de um item muda? O que vocês fariam a respeito?
- Como executamos o código que temos em `fury`?