IT BOARDING BOOTCAMP

Programación Java

Consumo de APIs - Uso de Servicios

// Práctica

Objetivo

El objetivo de esta práctica es el de ejercitar cómo consumir información de un microservicio dado.

¡Buena suerte! 😎👍

Ejercicio 1

La API de items es una de las APIs más importantes del marketplace de mercadolibre. Cada publicación que hace un vendedor de MeLi se obtiene desde esta API utilizando endpoints de la siguiente manera:
curl https://api.mercadolibre.com/items/<item_id>

Sabemos a su vez que cada publicación tiene un precio en una moneda específica, existe una API que nos resuelve la conversión entre monedas (link a la documentación pública)

Necesitamos entonces implementar una API que reciba un item ID, y vaya a buscar el precio en moneda original para ese item id, recupere la conversión con la moneda de origen, y devuelva el precio en dólares para ese ítem.

Nuestra API recibirá un GET a un endpoint /items/prices/[item-id] y deberá devolver

Tenemos entonces que responder las siguientes preguntas (en equipos):

- Cómo sería el flujo de ejecución desde que nos llega un request?
- Qué ocurriría si nos llegan 1000 requests por minuto? Y si llegaran 100.000? Esta API tiene una implementación correcta? Qué preguntas harías para pensar en optimizaciones a hacerle a la API
- Asumiendo que las cotizaciones cambian una vez al día. Qué opción podemos tomar para mejorar esta API?

Ejercicio 2 (Dia 2)

Partiendo del ejercicio anterior, tomar como base la implementación en el branch feature/add-currency-conversion-service de la aplicación bootcamp-demo-java-app para implementar las optimizaciones discutidas en la clase práctica del día 1

Parte 1 (en equipos de 6):

- Qué parte de la información podríamos tener pre-cacheada, dónde, sin tener pérdida de precisión en el cálculo?
- Cuál sería la ventana de exposición a tener un error de cálculo en el precio en USD? Cómo podemos ajustarla?
- Cómo implementarías esa cache?
- Qué servicios utilizarías para implementarlo?
- Indica qué clases modificarías para poder llevar adelante estos cambios

Demo time!!! Mostramos una implementación específica de KVS. Funcionamiento general del SDK, y la integración con un servicio específico que consume la información.

Parte 2 (en equipos de 6):

- Cómo harían para evitar ir “en vivo” a la API de items?
- Qué pasa si cambia el precio de un item?
- Cómo puedo enterarme inmediatamente cuando cambia el precio de un item? Qué harían al respecto?
- Cómo ejecuto el código que tengo en fury?