**Technical Test**

**JAVA DEVELOPER**

1. Create service for item data

2. Create service for transaction data

API Usage:

1. Get/Search item

2. For selected item, add item to cart

3. Add more items to cart

4. Remove item from cart

5. Finalize cart (calculcate price)

Optional:

1. Use spring cloud security

Untuk melakukan pengetesan diharapkan untuk import JSON ke postman terlebih dahulu untuk mempermudah pengetesan. Hal pertama yang harus dilakukan adalah memanggil API {{baseUrl}}/items/seed (baseUrl yang saya gunakan adalah <http://localhost:8000/>) atau menjalankan Set Seed yang sudah tersedia pada postman collection.

1. Get/Search item

Untuk melakukan perintah ini maka penguji dapat mencoba menjalankan Find item pada postman collection yang tersedia. Find item mengembalikan data semua item yang ada pada database. Find item juga bisa menyaring beberapa item berdasarkan atribut yang tersedia pada item. Saya menggunakan bantuan sipios untuk fitur filter/search tersebut. Untuk dokumentasi sipios dapat dilihat pada (https://github.com/sipios/spring-search). Selain itu saya menyediakan Get item untuk mengambil data satu item saja berdasarkan id item tersebut.

1. For selected item, add item to cart
2. Add more items to cart

Untuk poin kedua dan ketiga menurut saya hanya perlu 1 API yang sama tapi api tersebut harus bisa menerima banyak item sekaligus untuk disimpan. Penguji dapat mencoba menjalankan Save Cart Item pada postman collection yang tersedia. API yang saya buat akan mengembalikan data item yang disimpan beserta atribut tambahan seperti jumlah quantity pada cart dan jumlah harga sementara pada cart tersebut jika API berhasil dijalankan. API akan mengembalikan error jika quantity yang dimasukkan lebih banyak daripada quantity item yang tersedia pada database. API juga akan menambah quantity item yang sudah ada pada cart jika penguji memasukkan item yang sama berkali-kali.

Tambahan: saya menghardcode cart agar hanya tersimpan 1 record saja pada database, seharusnya cart pada e-commerce menurut saya tercipta Ketika customer mendaftar. Karena program saya tidak membuat system untuk customer maka saya membuat cart selalu mempunyai id 1 dan hanya ada 1 cart pada database.

1. Remove item from cart

Untuk melakukan perintah ini, maka penguji dapat mencoba menjalankan Delete all item from cart. Delete all item from cart berfungsi untuk mengosongkan cart dari semua item yang masuk kedalam cart tersebut. Saya juga membuat Delete one item from cart untuk menghapus satu item dari cart tersebut. Item tersebut akan dihilangkan tidak peduli berapa banyak jumlah item tersebut yang sudah dimasukkan ke cart. Saya buat demikian karena kalau user ingin mengurangi jumlah suatu item, maka API tersebut seharusnya merupakan API update bukan delete.

Saya juga membuat Get Cart Items untuk mempermudah pengecekan item-item apa saja yang tersimpan pada cart.

1. Finalize cart (calculcate price)

Untuk melakukan perintah ini, makan penguji dapat mencoba Checkout items from cart yang tersedia pada postman collection. API ini akan mengembalikan jumlah barang(semua item) yang dibeli, jumlah harga yang harus dibayar, item apa saja yang dibeli, jumlah item yang dibeli (peritem), dan harga item yang dibeli (per item). Setelah itu cart akan menjadi kosong dan quantity item pada database akan berkurang.