**Please Read the Below Instructions Carefully**:

Duration of this exercise is 60 minutes. This isn’t any stipulated duration; only a guidance.

* Make any necessary assumption, and clearly state the assumptions made.
* Do not seek any help – online or through any other source.
* **This Exercise is divided into 2 sections, Design section and Code section.**
* **Overall Duration of the exercise is 1 hr**.

Software suggestions

* There are 3 IDEs installed already on the system – Intellij, Eclipse and STS. Feel free to pick the one you like
* It’s recommended for speed that you use an in-memory database that comes with spring boot (type h2 in start.spring.io to generate starter for dependencies). You won’t have admin privileges to install any external database like oracle or MySQL.
* You can use any data-source if not familiar with above database e.g. files or utilities.
* In case you need any particular software or assistance, please contact HR
* The panel may be using this exercise for overall discussion. Please carry your m/c to panel.

Evaluation criteria

* Code Completeness/ Correctness
* Code Structure and quality: Modularity, usage of OO principles, size of classes/functions, class/function/variable names, package/class structure, java 8/9 features
* Choice of data structures and Design Patterns
* Unit Test cases using JUNIT (if you’re not conversant with JUNIT, just list down unit test cases)

**Please Note**:

1. **No Need to Implement UI Pages for this Exercise. Please define service layer.**
2. Make assumptions for any questions that you may have.
3. Please ensure an end to end implementation is done for given functions. You can pick min 2 operations to be implemented.

Problem Statement:

XYZ B2C ecommerce market place is working on building a solution to search an online product catalogue. Create an application to search product catalogue from its database. It’s desired for the customer to have a faster search along with consistent data across multiple online channels. The customer desires to have refreshed data every 2hrs based on any available inventory or change in any other parameters considered.

Product catalogue details:

A product (Shirts) can be from multiple brands (*Please assume your favorite brands*). There could be multiple products categories e.g. Pants, Socks, Caps, etc. The design should be able to accommodate different products categories.

Key search outcomes against the product:

The application should support the following operations

1. Group by brand / client
2. Group by price
3. Group by Color
4. Group by Size
5. Get by SKU
6. Available number of product by seller

Data Structure:

* Create database schema to store above values. Use in memory DB
* Data will come from multiple suppliers – this should trigger data refresh should happen on addition / deletion of brand or supplier
* Should support multiple channels channel for search – browser, mobile, kiosk to get a consistent output.

List various restful end points that can be developed

Non-functional requirements:

1. There will be multiple customers trying to access this search at the same time some of them may be buying the product as well.
2. All channels should return similar search output in terms of data and format.
3. You can assume it for a single Market but should be scalable to accommodate different market regions.

**Disclaimer:**

**This document is meant to assess your technical skills and is classified as "Sapient confidential". This document by any means shall not be used/shared without permission from Sapient, non-adherence to this can get your candidature blocked for employment with Sapient.   
  
In case you have seen this assessment exercise previously, please immediately inform your point of contact in Sapient and request for a similar different exercise for fair evaluation.**