Footway+ Challenge - Fullstack 2024

Thank you for taking the time to complete this coding challenge!

Footway+ is developing a crucial component of our Operations as a Service (OaaS) platform: a variant mapping system for multi-sales channels. This system will allow merchants to match their physical products (stored with Footway+) against digital products (listed on platforms like Shopify).

Note: We understand that this challenge covers a broad range of functionalities. You are not expected to complete all points within the given timeframe. Instead, focus on demonstrating your problem-solving approach, code quality, and architectural decisions. Choose the aspects you feel best showcase your abilities and explain your thought process for areas you didn't have time to implement fully.

System Requirements:

Backend(SpringBoot/Java):

- Design and implement a REST API that supports the following functionalities:
  - Managing physical and digital products (CRUD)
  - Creating and managing mappings between physical and digital products
  - Searching for products
  - Simulating order placement
- Your API should handle product attributes including SKU, EAN, name, description, price, and variants (like size/color)
- Implement proper error handling

Frontend(React/Typescript):

- Develop a user interface that allows users to:
  - View and manage physical and digital products
  - Create and edit mappings between physical and digital products
  - Search for products using free text search on product name/descriptions (for filtering product while creating mapping)
  - View and apply automatic mapping suggestions when EANs match
  - Manually override mapping suggestions
  - Filter products by mapping status (mapped, unmapped, suggested)
  - Place a simulated order and view the result

Data Storage:

- Choose and implement an appropriate data storage solution

Search and Matching Logic:

- Implement efficient search for free text search (Preferably at the backend/DB level)
- Create a robust matching suggestion system based on EAN

Scenario to understand the workflow:

let's say you want to open a shoe store and have shipped a bunch of shoes to the warehouse. After the warehouse processes them, they will be registered as physical products (the products stored in the warehouse under your inventory). Then later, when you want to sell them through your webshop (let's say Shopify), the products you created there and later synced to our system will be registered as digital products.

Later, once you get an order, for the warehouse to know which product to ship, it needs to know which physical products are mapped to which digital product, since they were notified on which digital product was ordered. So, a tool is needed for the store owner to map their physical products with the correlated digital products. Both digital products and physical products should have an optional field called EAN, which is the value of the barcode. If the EAN matches between a digital product and a physical product, it should be displayed as a suggestion.

In this challenge, it's sufficient to just have a simple endpoint to create digital and physical products and give people the possibility to view suggestions on potential mapping. If no suggestions are present, then allow users to map products by searching.

Technical Requirements:

- Design clean APIs with consistent naming conventions
- Implement efficient and scalable solutions, particularly for search and filtering
- Write clean, maintainable, and well-documented code
- Ensure proper separation of concerns
- Include basic unit tests for critical components
- Handle errors consistently and provide meaningful error messages to users

Optional Features (if time permits):

- Implement multi-sales channel functionality:
    - Allow a merchant to mark their product as "Multi-Sales Channel Enabled"
    - Allow other merchants to view and select products enabled for multi-sales channels
    - Enable cross-merchant mapping of physical to digital products
    - Implement a shipping priority system where a merchant's own products are shipped first
- Add basic authentication and authorization for merchant users
- Create a simple dashboard for mapping statistics or recent orders
- Implement performance optimizations for large datasets

When you're done:

- Publish the code to a private repo in Github and invite `nicksenap` at least 24 hours before your scheduled interview.