Development task

Setup

Imagine the following setup: We have a set of microservices running in a kubernetes cluster. In addition there is a postgreSQL database inside the same network segment (physically). Currently we have a service providing detailed information about a product. This service uses the postgreSQL database. The service is called by the frontend via an api gateway.

Goal

We want to implement a new service that uses some "AI" to enrich our product details with information gathered from other pages on the internet. The enriched data should be stored in a SOLR database that runs in another data center connected via VPN. The new service should replace the current service in the future.

Details

Data structures

The data structure currently used is quite simple:

example of current data structure

```json
{
    "sku": "article123",
    "detail": {
       "title": "best article ever",
       "description": "An astonishing article with unsurplussed features.",
       "price": 3.33,
       "currency": "USD"
    }
}
```

The enriched version is similar:

example of enriched data structure

```
{
    "sku": "article123",
    "detail": {
       "title": "best article ever",
       "description": "An astonishing article with unsurplussed features.",
       "price": 3.33,
       "currency": "USD",
       "averageprice": 3.28,
       "lowestprice": 3.00,
       "highestprice": 3.70,
       "medianprice": 3.30,
       "occurencecount": 23
    }
}
```

Endpoints

The current service has only one endpoint:

```
GET /api/product/detail/<sku>
```

returning the data structure given above.

The new service provides the same endpoint (this time returning the enriched data structure) plus an endpoint to push the current data to the service.

```
POST /api/product/add
```

with the current data structure given as body an HTTP 200 as result in case of success.

AI

The "AI" is provided by a not yet existing service with the endpoint "https://ai-service.manufactum.de/api/ai/<sku>" and returns just a JSON with the new values (average price, lowest price, highest price, median price and count).

Tasks

Implementation

Implement the part of the new service that provides the "POST" endpoint and stores the data in the

SOLR database.

You can "fake" the AI part: Instead of calling the endpoint just return the values.

- language: java or kotlin
- framework: spring-boot or micronaut

"Ticket"

What are the main aspects to consider while implementing the "GET" endpoint?