Java challenge

Part 1

Overview

You will be responsible to create a restful API that reads products' data from a mongo collection, then fetch prices for these products from an existing API and return the product's data with the **current** price.

Data sources

The mongodb is running at localhost, port 27018. The database name is code_challenge, the collection name is products.

The prices API is running at localhost, port 8081. The path is as follows:
basePath/api/prices/{product-code}, for example: http://localhost:8081/api/prices/1
This API returns a history of prices and some predictions of prices. In shorts, it will have timestamps with values. The timestamps can potentially be in the past, present and in the future.

You can find this api's documentation in http://localhost:8081/v3/api-docs

What to do

Using the predefined project provided to you (feel free to create any new files, packages, classes, configs, etc):

- The application should run in port 8080
- It should have a GET endpoint that returns a list of products with all the fields from the database and the **current** price
- Current price should be null in case no prices are found for the product
- The products list should be ordered by the product code ASC
- The endpoint path should be /page-api/products
- The endpoint should be paginated, with optional request params
  - page : default to 0
  - size : default to 20
- Service class unit test(s)
- Integration test of the endpoint

Not required but good to have :

- Enhanced performance with parallelism
- Extra unit tests
- Write logs during execution (log to console)
- Dockerfile

Part 2

The application contains a class named DataGenerator with a method which just returns a Map<Integer, String>.

There is a unit test written for the method, but the test is failing. Find out the reason for the failure and fix the DataGenerator method to pass it without changing the test itself.