Product aggregator microservice

Python 3 backend developer exercise

Create REST API JSON Python microservice, which allows to browse a product catalog and automatically updates prices from the offer service (provided by us, documentation below).

Requirements:

- Provide API to create, update and delete product
- Periodically query provided microservice for offers/shops with products

Data model:

Products - Each product corresponds to a real world product you can buy
id: The type is up to you :-)
name : string
description: string

Offers - Each offer represents a product offer being sold for some price somewhere
id: The type is up to you :-)
price : integer
items_in_stock: integer

Relations:

- **Product** has many **Offers**
- **Offer** belongs to **Product**

Specification:

Must have:

- Use **an SQL database** as internal database, library for API layer is up to you :-)
- Request an access token from the offers microservice
    - This should be done only once, all your registered products are tied to this token. Provide this token for all calls to the offers microservice.
- Create CRUD for **Products**
    - When a new **Product** is created, call the offers microservice to register it.
    - Your API does not need authentication
- Create background (job) service which periodically call the offers microservice to request new/updated offers for your products (price from offer ms is updated every minute).
- Base URL for the Offers MS should be configurable via an environment variable

- Write basic tests with pytest
- Push your code into a public repo on Github
- Add a README with information about how to start and use your service
- Send us a link

Nice to have:

- JSON REST API simple authentication (eq.: access-token)
- Consider to add some reasonable error handling to API layer
- Provide working Dockerfile and docker-compose.yml for your application for easy testing
- Use reasonable dependency management (requirements.txt, Pipenv, etc.)
- Deploy your application to Heroku
- Track history of offer price and create endpoint which returns offer price trend (array of prices) and compute the percentage rise/fall in price for a chosen period of time

Offers microservice - Documentation

Base URL: https://applifting-python-excercise-ms.herokuapp.com/api/v1

POST /auth

Headers: none
Request: none
Response:
201 CREATED
{
	“access_token”: “<uuid-token>”
}

POST /products/register

Headers: Bearer: <value>
Request:
{
	“id”: <id from your MS>
	“name”: “Benzinová sekačka Dosquarna”,
	“description: “Nejlepší sekačka na trhu. TLDR”
}
Response:
201 CREATED
{
	“id”: <id from your MS>
}
400 BAD REQUEST
{
	“code”: “BAD_REQUEST”,
	“msg”: <message>
}
401 UNAUTHORIZED
{
	“code”: “UNAUTHORIZED”,
	“msg”: <message>
}

GET /products/*:id:*/offers

Headers: Bearer: <value>
Request: id in path
Response:
200 OK
[
    {
        “id”: <id from offer service>,
        “price”: 1000,
        “items_in_stock: 5
    }
]
400 BAD REQUEST
{
    “code”: “BAD_REQUEST”,
    “msg”: <message>
}
401 UNAUTHORIZED
{
    “code”: “UNAUTHORIZED”,
    “msg”: <message>
}
404 NOT FOUND
{
    “code”: “NOT FOUND”,
    “msg”: <message>
}