Python 3 BE developer task

Product aggregator microservice

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

Requirements:

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

Data model:

Products: each product corresponds to a real world product you can buy
- **id:** any type of your own choosing
- **name:** string
- **description:** string

(*a **Product** has many **Offers***)

Offers: each offer represents a product offer being sold for a price
- **id:** any type of your own choosing
- **price:** integer
- **items_in_stock:** integer

*(an **Offer** belongs to a **Product**)*

Specification:

- **Must-haves:**
    - Use an **SQL database** as an internal database; the library for the 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**
    - Once a new **Product** is created, call the **Offers** microservice to register it
    - Your API does not need authentication
    - Create a background (job) service which periodically calls the **Offers** microservice to request new/updated offers for your products (*price from offer ms is updated every minute*)
    - Create a read-only API for product offers
    - 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** (*and then send us a link*)
    - Add a **README** with information on how to start and use your service

applifting.cz
info@applifting.cz

Filip Kirschner | COO
+420 739 627 797

What you can earn some "extra points" for:

- JSON REST API simple authentication (*eq.: access-token*)
- Consider adding some reasonable error handling to the API layer
- Provide a 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 the history of offer prices and create an endpoint which returns the trend in offer prices *(array of prices*) and compute the percentual 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>
      }
      ```

applifting.cz
info@applifting.cz

Filip Kirschner | COO
+420 739 627 797

Applifting

- 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>
}
```

Alright, that’s it. Good luck and, most importantly, **have fun!**

applifting.cz
info@applifting.cz

Filip Kirschner | COO
+420 739 627 797