Fonoma – Backend Developer Test

Complete this simple project to proceed with the hiring process.

We will review your implementation and provide feedback based on the quality of your solution.

Good luck!

Instructions

1. Create a [FastAPI](https://fastapi.tiangolo.com/) (https://fastapi.tiangolo.com/) basic project following this tutorial https://fastapi.tiangolo.com/tutorial/first-steps/
2. Create a git repository with the code and upload it to Github, Gitlab, or Bitbucket.
3. In the repository, create a new branch.
4. In the new branch, implement a new endpoint with the path /solution. This endpoint should accept a POST request with the parameters of the coding exercise described in the section **Coding exercise** that is below. The endpoint should return the result of executing the function process_orders with the parameters obtained from the post.

   Sample request:

   Bash:
   ```bash
   curl -X POST "https://your-app-url." -H "Content-Type: application/json"
               -d '{"orders": [{"id": 1, "item": "Laptop", "quantity": 1, "price": 999.99, "status": "completed"},
               {"id": 2, "item": "Smartphone", "quantity": 2, "price": 499.95, "status": "pending"},
               {"id": 3, "item": "Headphones", "quantity": 3, "price": 99.90, "status": "completed"},
               {"id": 4, "item": "Mouse", "quantity": 4, "price": 24.99, "status": "canceled"},], "criterion": "completed"}'
   ```

   Response: `1299.69`

IMPORTANT: The code of the endpoint should validate the input parameters. For example, it should check that the price of an item is not negative.

5. Write at least one unit test for the /solution endpoint. More tests will give you bonus points.
6. Create a Pull Request (or Merge Request) of the new branch with a detailed description of the changes made.
7. Deploy the application to [Render.com](https://render.com/) (https://render.com/). See this example https://github.com/render-examples/fastapi.
8. After completing the test and deploying the application, send us an email with the following links:
   - The link to the PR in your repository.
   - The link to the deployed version on [Render.com](https://render.com/) (https://render.com/).

Bonus points: In step 4, use a docker image for the deployment. See https://render.com/docs/docker#getting-started-with-docker.
Bonus points: Use Redis to cache the results of the requests to /solution.
Bonus points: Add type annotations.

Coding exercise

Implement a single function named process_orders that receives a list of orders and a filter criterion. The function should filter the orders based on the criterion and return the total revenue for the filtered orders. The function signature should be as follows:

Python
```python
def process_orders(orders, criterion):
    pass
```

The input parameter orders is a list of dictionaries, where each dictionary represents an order with the following keys:

- id: an integer representing the order ID
- item: a string representing the item name
- quantity: an integer representing the number of items in the order
- price: a number representing the price per item
- status: a string representing the order status, which can be either completed, pending, or canceled

The criterion is a string that indicates the filter to be applied to the orders. The function should support the following criteria:

- completed: Only consider orders with the status completed.
- pending: Only consider orders with the status pending.
- canceled: Only consider orders with the status canceled.
- all: Consider all orders, regardless of their status.