FlyFlat Senior Python Developer Interview Exercise

Overview

Thank you for your interest in the Senior Python Developer position at FlyFlat. As part of our interview process, we'd like you to complete a coding exercise that demonstrates your skills in Python web development, focusing on Django and FastAPI integration, API design, and asynchronous processing.

Your task is to create a Flight Data Enrichment System that processes flight data, enriches it with pricing information, and presents the results. This exercise will test your ability to work with both Django and FastAPI, implement task queues, integrate external APIs, and design efficient database schemas.

Requirements

1. **FastAPI Component**

    - Create an endpoint that accepts flight data as a payload (see example below).
    - Implement a task queue system (e.g., Celery) to process the enrichment requests asynchronously.
    - Create an endpoint to check the status of a task given a task ID.
    - When the task is complete, return abridged flight info, request status, and retail price (if successful).

2. **Django Component**

    - Implement a view displaying the task queue, showing the last 3 completed tasks and their result status.
    - Create a view that provides a table of flights with key columns only.

3. **Data Enrichment**

    - Use SerpAPI to fetch the retail price for the flight.
    - Handle potential errors or rate limiting from the external API.

4. **Database**

    - Design an appropriate database schema to store flight information and task results.

5. **Testing**

- Write a simple test that tests your endpoint functionality using your favorite Python test suite (pytest, unittest, etc.).

6. **Documentation**

    - Provide clear instructions on how to set up and run your project.
    - Document any assumptions or design decisions you've made.

Evaluation Criteria

We will assess your submission based on:

1. **Code quality and organization**
2. **Proper use of Django and FastAPI features**
3. **Effective implementation of the task queue system**
4. **Error handling and edge case consideration**
5. **Database design and query optimization**
6. **Quality of the written test**
7. **Documentation clarity**

Submission Guidelines

- Submit your project as a **private** GitHub repository.
    - Please add **@berubejd** and **@DocisJP** to your GitHub repository.
- Include a comprehensive README.md with:
    - Setup instructions
    - Architecture decisions and rationale
    - Feature implementation details
    - Known limitations and future improvements
- Provide any necessary requirements files (e.g., requirements.txt).

Input Data Example

```
{
  "id": "20250613-MS-MS986-MS747",
  "travel_class": "Business",
  "origin": "JFK",
  "destination": "ATH",
  "departure_time": "2025-06-13 12:55",
  "arrival_time": "2025-06-14 12:10",
  "flight_numbers": [
    "MS986",
```

```json
    "MS747"
  ],
  "legs": [
    {
      "origin": "JFK",
      "destination": "CAI",
      "departure_time": "2025-06-13 12:55:00",
      "arrival_time": "2025-06-14 06:15:00",
      "flight_number": "MS986",
      "aircraft_type": "Boeing 777",
      "cabin_type": "Business",
      "duration": 620,
      "layover_time": "235.0",
      "distance": 5614
    },
    {
      "origin": "CAI",
      "destination": "ATH",
      "departure_time": "2025-06-14 10:10:00",
      "arrival_time": "2025-06-14 12:10:00",
      "flight_number": "MS747",
      "aircraft_type": "Boeing 737",
      "cabin_type": "Business",
      "duration": 120,
      "layover_time": "0.0",
      "distance": 688
    }
  ],
  "last_seen": "2025-05-29T03:38:05Z"
}
```

Follow-up Discussion

After reviewing your submission, we will invite you to a follow-up discussion with our team. This session will provide you with the opportunity to:

1. Walk us through your implementation and highlight key decisions you made during the process.
2. Explain your approach to problem-solving and any challenges you encountered.
3. Discuss and implement potential improvements or alternative solutions you considered.
4. Showcase your ability to communicate technical concepts effectively.

We value this discussion as it allows us to gain deeper insights into your thought process, problem-solving skills, and how you approach software development challenges.

Time Allocation

You have up to 2 hours to complete this exercise. This time does not need to be contiguous and can be spread across multiple days, if needed. If this exercise takes more than 2 hours, please either stop at that point or make a note of the total time spent

Questions

If you have any questions or need clarification on any aspect of this exercise, please don't hesitate to reach out to Jeff Berube.

Good luck, and we look forward to seeing your implementation!