Code Challenge: Hotel Review Search and Sorting

To work for Expedia, a Travel e-commerce company, we would like you to start by doing a code challenge test close to our business. You have been tasked with implementing a backend API for a hotel review search and sorting feature. Your goal is to create an API endpoint that accepts a search query, retrieves a list of hotels and their reviews from a datasource, calculates the average review score for each hotel, and orders the list of hotels based on their review average.

User Story

As a backend engineer, I want to be able to search for a list of hotels in a particular destination and then be able to order them based on their review average, so that users can easily find the most highly rated hotels.

To achieve this, I would create an API endpoint that accepts a search query as input, retrieve a list of hotels that match the search criteria from the datasource, retrieve the associated reviews for each hotel, calculate the average review score for each hotel, and order the list of hotels based on their review average.

Your API should have the following specifications:

Input

The API should accept a search query as input, which can include any combination of the following parameters:

- location (string): the location of the hotel
- checkin_date (date): the check-in date for the hotel
- checkout_date (date): the check-out date for the hotel
- price_range (array of integers): the price range for the hotel (e.g. [50, 100] for hotels that cost between 50 and 100 dollars per night)

Output

The API should return a list of hotels and their associated reviews, ordered by the review average. Each hotel object should include the following fields:

- id (string): the unique identifier for the hotel
- name (string): the name of the hotel
- description (string): the description of the hotel
- location (object): the location of the hotel
    - id (string): unique identifier for the location
    - name (string): the name for the destination
- total_price (integer): the total price for the stay
- image (string): the URL of the image representing the hotel
- reviews (array of objects): the list of reviews for the hotel, where each review object includes the following fields:
    - id (string): the unique identifier for the review

    - rating (integer): the rating for the review (out of 5 stars)
    - comment (string): the comment for the review

The list of hotels should be ordered by the review average, with the highest rated hotels appearing first.

Constraints

- The API can be implemented in Java, Kotlin or Scala.
- You can assume that the hotels and reviews are stored in its own datasource.
- You can use any libraries or frameworks that you deem appropriate.
- You should write tests for your implementation.

Deliverables

A git repository** (zip/tar file) containing the following:

- An implementation of the API endpoint that meets the above specifications.
- Tests that cover the expected behavior of the API.
- Any documentation or instructions for running your code and understanding your design decisions.

** We are requesting a git repository since it allow us to look into the change history and better understand how your code has evolved.

Expectations

As a requirement for this code challenge, we have established a set of guidelines that we expect your solution to adhere to:

- Each component in the application has clear responsibilities and can be easily replaced with different implementations.
- We look for simplicity in the code, don't overcomplicate it.
- Components should be *properly* tested.
- We may need you to add new features or improve existing ones later on. To ensure flexibility and scalability, please follow the above guidelines.

Note that this coding exercise excludes design decisions, which may cause ambiguity in your solution. If you face uncertainty, use your best judgment and document your reasoning for us to understand your thought process. Our aim is to gain insight into your problem-solving skills and coding approach.