

Caselet: Building an API Core Project for a Hotel Booking System

Client: XYZ Hotels

Overview: XYZ Hotels is a renowned hotel chain operating globally. To streamline their reservation process and provide an efficient booking experience to their customers, XYZ Hotels plans to develop an API Core Project using a code-first approach. The project should include CRUD operations, filtering capabilities, count functionality, JWT token authentication, and handle the one-to-many relationship between hotels and rooms.

Challenge: XYZ Hotels needs a comprehensive API Core Project that enables customers to make reservations, hotel staff to manage room availability, and provides secure access through JWT token authentication. Additionally, the project should handle the one-to-many relationship between hotels and rooms, where each hotel can have multiple rooms.

Objectives:

1. **CRUD Operations:** Develop APIs to support CRUD operations for managing hotel information, including creating new hotels, updating hotel details, retrieving hotel information, and deleting hotels.
2. **Filtering:** Implement filtering capabilities to allow customers to search and filter hotels based on criteria such as location, price range, or amenities.
3. **Count Functionality:** Enable users to obtain counts of available rooms in specific hotels, providing insights into room availability for better decision-making.
4. **JWT Token Authentication:** Implement a secure authentication mechanism using JWT tokens to ensure that only authorized users can access the API endpoints, safeguarding customer and hotel data.
5. **One-to-Many Relationship:** Establish an efficient solution to handle the one-to-many relationship between hotels and rooms, where each hotel can have multiple rooms.
6. **Exception Handling:** Implement try-catch blocks to handle exceptions gracefully, providing meaningful error messages and ensuring the system's stability.
7. **Repository Pattern:** Apply the repository pattern to separate the data access layer from the business logic, promoting code modularity and maintainability.

Approach:

1. **Database Design:** Define the database schema, including tables for hotels, rooms, and the necessary foreign key relationships to handle the one-to-many relationship.
2. **Code-First Development:** Employ a code-first approach to generate the database schema based on the defined models and relationships, utilizing frameworks like Entity Framework.
3. **Repository Implementation:** Implement repositories for hotels and rooms, following the repository pattern to handle data access operations and encapsulate data persistence logic.
4. **CRUD Operations:** Implement APIs to support creating, reading, updating, and deleting hotel information, allowing hotel staff to manage hotel details effectively.
5. **Filtering:** Develop APIs that enable customers to search and filter hotels based on specified criteria, such as location, price range, or amenities, to facilitate the booking process.
6. **Count Functionality:** Implement APIs to provide counts of available rooms in specific hotels, allowing users to retrieve the number of available rooms in each hotel for accurate room availability information.

7. **JWT Token Authentication:** Integrate JWT token authentication to secure the API endpoints, ensuring that only authenticated users, such as registered customers and authorized hotel staff, can access the protected resources.
8. **Exception Handling:** Implement try-catch blocks to handle exceptions gracefully, capturing specific exceptions, logging the errors, and providing meaningful error responses to the API consumers.
9. **One-to-Many Relationship:** Design and implement the necessary database mappings and APIs to handle the one-to-many relationship between hotels and rooms, allowing each hotel to have multiple rooms and facilitating efficient room management.

By successfully implementing this API Core Project, XYZ Hotels will have an advanced reservation system that allows customers to search and book hotels easily, hotel staff to manage room availability effectively, and provides secure access through JWT token authentication. The project will handle the one-to-many relationship between hotels and rooms seamlessly, ensure accurate room availability information, and maintain code modularity and stability through the implementation