DirectBookingOne.Api

Hotel Product Management with MongoDB

This document provides a solution overview for managing hotel product categories and their availability using ASP.NET Core, C#, and MongoDB.

1. Functionality Overview:

The API offers functionalities to manage hotel product categories and check their availability:

- **Product Management:**
  - List all available product categories.
  - Retrieve a specific product category by its ID.
  - Create a new product category.
  - Update an existing product category.
  - Delete a product category.
- **Availability Check:**
  - Check if a specific product category is available for a given date range.

2. Technology Stack:

- **Backend:** ASP.NET Core API
- **Programming Language:** C#
- **Database:** MongoDB

3. Solution:

Root Directory:

- **appsettings.json:** Stores configuration settings for the application, such as connection strings to MongoDB and logging levels.

Models Folder:

- Contains class definitions for modeling data entities:
  - `BaseProduct.cs` (likely): An abstract base class defining common properties inherited by other models (e.g., ID, CreatedAt).
  - `Product.cs`: Represents a product category with properties like CategoryName, Capacity, and PricePerNight.
  - `Availability.cs`: Stores availability information for a product, including ProductId, StartDate, and EndDate.

Services Folder:

- Holds service classes for interacting with the database:
    - `ProductService.cs`: Handles interactions with the product collection in MongoDB (CRUD operations for creating, reading, updating, and deleting products).
    - `AvailabilityService.cs`: Provides methods for checking availability for a product based on existing availability records and creating new availability entries.

Controllers Folder:

- Implements API endpoints using ASP.NET Core MVC controllers:
    - `ProductController.cs`: Implements endpoints for managing product categories using the `ProductService`.
    - `AvailabilityController.cs`: Implements the availability check endpoint using the `AvailabilityService`.

Other Files:

- **Program.cs:** The application entry point responsible for starting the ASP.NET Core web application.

This structure promotes separation of concerns:

- Models define data entities.
- Services handle data access logic.
- Controllers expose API endpoints using services.

4. Conclusion:

This walkthrough showcases a solution for managing hotel product categories and their availability. The code leverages ASP.NET Core, C#, and MongoDB, demonstrating core functionalities and adherence to best practices.