Session 4 HW - HW3 Enhancements

Last edited time	@July 9, 2024 5:59 PM

Homework: Enhancing the Movie Management API with MongoDB

Objective

Expand your existing Movie Management API by integrating MongoDB for robust data handling. This assignment emphasizes creating a data access layer (DAL) that manages all database interactions using `MongoClient`, separating concerns for improved maintenance and scalability.

Requirements

- An existing Movie Management API project developed using Express.js (without database integration).
- Node.js and npm installed.
- MongoDB installed locally or accessible via MongoDB Atlas.
- Basic understanding of MongoDB operations and `MongoClient`.

Focus: Database Layer Development

The primary task is to develop the database interaction components of your application using MongoDB. This involves setting up models, services, and a DAL to handle all database operations.

Setup Instructions

1. **Prepare Your Environment:**
    - Ensure your MongoDB instance is ready and accessible.
    - In your existing project, install the MongoDB Node.js Driver if not already installed: `npm install mongodb`.

2. **Database Connection:**

    - Configure `MongoClient` to connect to your MongoDB database in a separate configuration file or as part of your main server file, depending on your project structure.

Task Breakdown

1. **Models:**

    - Modify the `Movie` model to interact with MongoDB. Define the schema directly in MongoDB terms using the capabilities of `MongoClient`.

2. **Data Access Layer (DAL):**

    - Create a `MovieRepository` class in the DAL folder that encapsulates all MongoDB operations. This class should include methods for each CRUD operation.

    - Methods should include:

        - `findAll()`: Retrieves all movie entries from the database.
        - `findById(id)`: Retrieves a single movie by its MongoDB ID.
        - `create(movie)`: Inserts a new movie into the database.
        - `updateById(id, movie)`: Updates an existing movie record.
        - `deleteById(id)`: Removes a movie record from the database.

3. **Services:**

    - Ensure that your existing `MovieService` interacts with the `MovieRepository` for all data operations, abstracting away the direct database interactions from the service logic.

4. **Integration:**

    - Modify any existing API endpoints to use the new DAL methods for data operations, ensuring that the controllers now interact with the service layer which in turn uses the DAL.

Deliverables

- Updated source code for the Movie Management API with a fully integrated MongoDB database layer.
- A brief report documenting the changes made to the project structure, particularly focusing on the database integration and any challenges encountered.