**Item Management API - Hometask by Connect - README**

This API allows users to manage items and categories. It includes endpoints for adding, updating, searching, and retrieving items and categories.

**Setup Instructions**

**Step 1: Download & Navigate to Project Folder**

Navigate to the project folder via terminal:

cd <project\_folder>

**Step 2: Install Dependencies**

In the project folder, run the following command to install the required Node.js packages. The packages used include **dotenv**, **MySQL2**, **joi**, **Express**, and **Sequelize**.

npm install

**Step 3: Set Up MySQL Database with XAMPP**

1. **Start XAMPP**: Open XAMPP and start the **Apache** and **MySQL** modules.
2. **Access phpMyAdmin**:
   * Open a browser and go to http://localhost/phpmyadmin.
3. **Create a Database**:
   * In phpMyAdmin, create a new database named connect\_api.
4. **Create Tables Using Sequelize**:
   * Instead of manually creating tables, use Sequelize to handle migrations. Run the following commands to automatically create the tables and populate mock data:

**Run Migrations**

This command creates the tables in the database based on the models and migration files:

npx sequelize-cli db:migrate

**Run Seeders**

This command inserts mock data into the tables (10 items and 2 categories as per requirements):

npx sequelize-cli db:seed:all

**Step 4: Configure Environment Variables**

In the project folder, update the .env file according to your database configuration. Below is an example of the necessary variables:

DB\_HOST=localhost

DB\_USERNAME=root

DB\_PASSWORD=yourpassword

DB\_DATABASE=connect\_api

DB\_DIALECT=mysql

API\_KEY=your\_api\_key\_here

Replace yourpassword with the MySQL root password (if using XAMPP with no password, leave it blank). Also, replace your\_api\_key\_here with a secure API key of your choice, or use the one I already provided in the existing .env file.

**Step 5: Run the Server**

Start the server by running:

node app.js

You should see a message indicating that the server is running on port 3000.

**Test the API**

The API includes the following endpoints:

* **POST /api/category** - Add a new category
* **GET /api/category/** - Get category details and associated items by category ID
* **POST /api/items** - Create or update an item with volumes
* **GET /api/items** - Retrieve all items
* **GET /api/item/** - Retrieve item details by ID
* **GET /api/item/search?query=<string>** - Search for items and categories based on a search string

**API Key Authentication**

All endpoints require an API key for authentication. Include the key in the apikey header for each request:

apikey: your\_api\_key\_here

Thank you for your time :)