Fantasy RPG Inventory Server

This project is a simple backend server for managing a Fantasy RPG inventory. It allows users to add, update, delete, and view RPG character items through a RESTful API.

Project Structure

This section outlines the structure of the `fantasy-rpg` project and provides a brief description of each file and directory:

```
fantasy-rpg/
│
├── middleware/
│   ├── db.js
│   ├── db.json
│   └── default_db.json
│
├── public/
│   ├── css/
│   │   ├── api.css
│   │   └── style.css
│   ├── img/
│   ├── js/
│   │   ├── index.js
│   │   └── toaster.js
│   ├── api.html
│   └── index.html
│
├── package.json
├── readme.md
└── server.js
```

`fantasy-rpg/`

Root directory of the Fantasy RPG project.

`middleware/`

Contains middleware and utility files related to data handling and processing.

- **`db.js`**: A JavaScript file containing logic to manipulate the `db.json` data, such as loading and saving data.
- **`db.json`**: The current state of the server's data, including characters, items, etc. This file acts as a simple database.
- **`default_db.json`**: A template or default state of the server's data. Used to reset `db.json` to its original state.

`public/`

Houses all static files served directly to the client, such as HTML, CSS, JS, and images.

- `css/`: Contains CSS stylesheets for the project.
  - `api.css`: Specific styles for the API documentation page (`api.html`).
  - `style.css`: General styles applied across the project.
- `img/`: Directory for storing image files used in the project.
- `js/`: Contains JavaScript files.
  - `index.js`: The main JavaScript file for the front-end logic, handling interactions in `index.html`.
  - `toaster.js`: A utility script for displaying toast notifications on the site.
- `api.html`: The API documentation page, explaining how to use the server's endpoints.
- `index.html`: The main entry point for the application's front-end, providing the user interface.

`package.json`

Describes the project's metadata, dependencies, and scripts. This file is used by npm to manage the project's packages.

`readme.md`

A Markdown file containing information about the project, including setup instructions, usage details, and documentation.

`server.js`

The main server file for the Fantasy RPG project. It initializes the Express server and defines the API endpoints.

Getting Started

These instructions will get you a copy of the project up and running on your local machine for development and testing purposes.

Prerequisites

Before you begin, ensure you have Node.js installed on your machine. This project was built with Node.js version `20.9.0`, so it is recommended to use a compatible version.

Installing

Follow these steps to get your development environment running:

1. **Extract the server file**

   ```
   unzip a02Server.zip
   ```

2. **Install dependencies**: Run the following command in the root directory of your project to install the necessary dependencies:

   ```
   npm install
   ```

3. **Start the server**: To start the server, run:

   ```
   npm run server
   ```

This command will start the server on http://localhost:4042 (or whatever port is specified in your environment variables). You can access the API through this URL.

Usage

Once the server is running, you can use the provided endpoints to manage the RPG inventory. Here are some examples of how to interact with the server:

- Get all items: Sends a **GET** request to **/api/items**
- Get a specific item: Sends a **GET** request to **/api/items/:id**
- Add a new item: Sends a **POST** request to **/api/items** (with item data as JSON).
- Update an item: Sends a **PUT** request to **/api/items/:id** (with the updated data as JSON).
- Delete an item: Sends a **DELETE** request to **/api/items/:id**

Visit http://localhost:4042/api for more details.

Demonstration

You can watch a video showing what your assignment is supposed to do:

Watch the Video.