Product & Category API Documentation

This document provides details about the API endpoints for managing Products and Categories.

Base URL: `http://localhost:3000` (during development)

Getting Started

Prerequisites

- Node.js (v14 or later recommended)
- npm (comes with Node.js)
- Python 3.x (for running tests)

Installation

1. Clone the repository
2. Install Node.js dependencies:

   ```
   npm install
   ```

3. Start the development server:

   ```
   npm run dev
   ```

   The server will run on `http://localhost:3000`

Running Tests

To run the Python integration tests:

```
python product_api_test.py
```

Development

- The server will automatically restart when files change (using nodemon)
- Database is SQLite and stored in `database.sqlite` at the project root

- In test environment, an in-memory database is used

Authentication

No authentication is required for these endpoints.

Common Error Responses

- **400 Bad Request:** Usually indicates a validation error (e.g., missing required fields, invalid data format, invalid ID format) or a non-existent referenced resource (like `category_id`). The response body typically contains an `error` message and sometimes a `details` array.

```
{
  "error": "Validation Error",
  "details": ["Product name cannot be empty"]
}
```

```
{
  "error": "Category with ID 999 does not exist."
}
```

```
{
 "error": "Invalid ID format"
}
```

- **404 Not Found:** The requested resource (e.g., product or category with a specific ID) could not be found.

```
{
  "error": "Product not found"
}
```

- **409 Conflict:** Attempting to delete a resource that cannot be deleted due to constraints (e.g., deleting a Category that still has associated Products).

```
{
   "error": "Cannot delete category as it is associated with existing products."
}
```

- **500 Internal Server Error:** An unexpected error occurred on the server.

```
{
  "error": "Server Error"
}
```

Categories API

Endpoints for managing product categories.

1. Create Category

- **Endpoint:** `POST /categories`
- **Description:** Creates a new category.
- **Request Body:** `application/json`

```
{
  "name": "string (required)"
}
```

Example:

```
{
  "name": "Electronics"
}
```

- **Success Response (201 Created):** `application/json` - The created category object.

```
{
  "id": 1,
  "name": "Electronics",
  "createdAt": "2023-10-27T10:00:00.000Z",
  "updatedAt": "2023-10-27T10:00:00.000Z"
}
```

- **Error Responses:** 400, 500

2. Get All Categories

- **Endpoint:** `GET /categories`

- **Description:** Retrieves a list of all categories.
- **Request Body:** None
- **Success Response (200 OK):** `application/json` - An array of category objects.

```
[
  {
    "id": 1,
    "name": "Electronics",
    "createdAt": "2023-10-27T10:00:00.000Z",
    "updatedAt": "2023-10-27T10:00:00.000Z"
  },
  {
    "id": 2,
    "name": "Books",
    "createdAt": "2023-10-27T10:05:00.000Z",
    "updatedAt": "2023-10-27T10:05:00.000Z"
  }
]
```

- **Error Responses:** 500

3. Get Category by ID

- **Endpoint:** `GET /categories/{id}`
- **Description:** Retrieves a single category by its numeric ID.
- **Path Parameter:**
  - `id` (integer, required): The ID of the category to retrieve.
- **Request Body:** None
- **Success Response (200 OK):** `application/json` - The requested category object.

```
{
  "id": 1,
  "name": "Electronics",
  "createdAt": "2023-10-27T10:00:00.000Z",
  "updatedAt": "2023-10-27T10:00:00.000Z"
}
```

- **Error Responses:** 400 (Invalid ID), 404, 500

4. Update Category

- **Endpoint:** `PUT /categories/{id}`

- **Description:** Updates an existing category by its numeric ID.
- **Path Parameter:**
  - `id` (integer, required): The ID of the category to update.
- **Request Body:** `application/json`

```
{
  "name": "string (required)"
}
```

Example:

```
{
  "name": "Consumer Electronics"
}
```

- **Success Response (200 OK):** `application/json` - The updated category object.

```
{
  "id": 1,
  "name": "Consumer Electronics",
  "createdAt": "2023-10-27T10:00:00.000Z",
  "updatedAt": "2023-10-27T10:15:00.000Z" // Note updated timestamp
}
```

- **Error Responses:** 400 (Validation or Invalid ID), 404, 500

5. Delete Category

- **Endpoint:** `DELETE /categories/{id}`
- **Description:** Deletes a category by its numeric ID. Fails if the category is associated with products.
- **Path Parameter:**
  - `id` (integer, required): The ID of the category to delete.
- **Request Body:** None
- **Success Response (200 OK):** `application/json`

```
{
  "message": "Category deleted successfully"
}
```

- **Error Responses:** 400 (Invalid ID), 404, 409 (Conflict), 500

Products API

Endpoints for managing products.

1. Create Product

- **Endpoint:** `POST /products`
- **Description:** Creates a new product. The `category_id` must refer to an existing category.
- **Request Body:** `application/json`

```
{
  "name": "string (required)",
  "desc": "string (required)",
  "image": "string (optional, nullable)",
  "category_id": "integer (required)",
  "price": "integer (required, non-negative)"
}
```

Example:

```
{
  "name": "Wireless Mouse",
  "desc": "Ergonomic wireless optical mouse",
  "image": "http://example.com/mouse.jpg",
  "category_id": 1,
  "price": 25
}
```

- **Success Response (201 Created):** `application/json` - The created product object.

```
{
  "id": 101,
  "name": "Wireless Mouse",
  "desc": "Ergonomic wireless optical mouse",
  "image": "http://example.com/mouse.jpg",
  "category_id": 1,
  "price": 25,
  "createdAt": "2023-10-27T11:00:00.000Z",
  "updatedAt": "2023-10-27T11:00:00.000Z"
}
```

- **Error Responses:** 400 (Validation or Invalid Category ID), 500

2. Get All Products

- **Endpoint:** `GET /products`
- **Description:** Retrieves a list of all products, including their associated category details.
- **Request Body:** None
- **Success Response (200 OK):** `application/json` - An array of product objects, each including a `category` object.

```
[
  {
    "id": 101,
    "name": "Wireless Mouse",
    "desc": "Ergonomic wireless optical mouse",
    "image": "http://example.com/mouse.jpg",
    "category_id": 1,
    "price": 25,
    "createdAt": "2023-10-27T11:00:00.000Z",
    "updatedAt": "2023-10-27T11:00:00.000Z",
    "category": {
      "id": 1,
      "name": "Electronics",
       "createdAt": "2023-10-27T10:00:00.000Z",
       "updatedAt": "2023-10-27T10:15:00.000Z"
    }
  },
  // ... other products
]
```

- **Error Responses:** 500

3. Get Product by ID

- **Endpoint:** `GET /products/{id}`
- **Description:** Retrieves a single product by its numeric ID, including its associated category details.
- **Path Parameter:**
  - `id` (integer, required): The ID of the product to retrieve.
- **Request Body:** None

- **Success Response (200 OK):** `application/json` - The requested product object, including a `category` object.

```
{
  "id": 101,
  "name": "Wireless Mouse",
  "desc": "Ergonomic wireless optical mouse",
  "image": "http://example.com/mouse.jpg",
  "category_id": 1,
  "price": 25,
  "createdAt": "2023-10-27T11:00:00.000Z",
  "updatedAt": "2023-10-27T11:00:00.000Z",
  "category": {
    "id": 1,
    "name": "Electronics",
    "createdAt": "2023-10-27T10:00:00.000Z",
    "updatedAt": "2023-10-27T10:15:00.000Z"
  }
}
```

- **Error Responses:** 400 (Invalid ID), 404, 500

4. Update Product

- **Endpoint:** `PUT /products/{id}`
- **Description:** Updates an existing product by its numeric ID. If `category_id` is provided, it must refer to an existing category.
- **Path Parameter:**
  - `id` (integer, required): The ID of the product to update.
- **Request Body:** `application/json` - Include only the fields to update. All fields are optional, but at least one must be provided for the update to potentially change anything.

```
{
  "name": "string",
  "desc": "string",
  "image": "string | null",
  "category_id": "integer",
  "price": "integer (non-negative)"
}
```

Example:

```
{
  "price": 30,
  "image": null
}
```

- **Success Response (200 OK):** `application/json` - The *entire* updated product object, including its category.

```
{
  "id": 101,
  "name": "Wireless Mouse", // Unchanged in example
  "desc": "Ergonomic wireless optical mouse", // Unchanged in example
  "image": null, // Updated in example
  "category_id": 1, // Unchanged in example
  "price": 30, // Updated in example
  "createdAt": "2023-10-27T11:00:00.000Z",
  "updatedAt": "2023-10-27T11:20:00.000Z", // Note updated timestamp
  "category": { // Included category info
    "id": 1,
    "name": "Electronics",
    "createdAt": "2023-10-27T10:00:00.000Z",
    "updatedAt": "2023-10-27T10:15:00.000Z"
  }
}
```

- **Error Responses:** 400 (Validation, Invalid ID or Invalid Category ID), 404, 500

5. Delete Product

- **Endpoint:** `DELETE /products/{id}`
- **Description:** Deletes a product by its numeric ID.
- **Path Parameter:**
  - `id` (integer, required): The ID of the product to delete.
- **Request Body:** None
- **Success Response (200 OK):** `application/json`

```
{
  "message": "Product deleted successfully"
}
```

- **Error Responses:** 400 (Invalid ID), 404, 500