**Product API Documentation**

The Product API allows you to perform various operations related to products, such as creating, retrieving, updating, and deleting products.

**How to Run Locally**

To run the server locally, follow these steps:

1. Make sure you have Node.js and npm installed on your machine.
2. Clone the repository.
3. Navigate to the project folder in the terminal.
4. Run the following commands:

**npm install**

**npm start**

The server will be running at [**http://localhost:3000**](http://localhost:3000).

**Base URL**

http://localhost:3000/api/products

**Endpoints**

**1. Get Product by ID:**

**Endpoint:** **GET /api/products/:id**

Retrieve details of a specific product by providing the product ID.

Parameters

* **id** (string): The unique identifier of the product.

**Example**

GET http://localhost:3000/api/products/1

**Response**

{ "id": "1", "name": "Product Name", "price": 19.99 }

**2. Get List of Products**

**Endpoint:** **GET /api/products**

Retrieve a list of all products.

**Example**

GET http://localhost:3000/api/products

**Response**

[ { "id": "1", "name": "Product 1", "price": 19.99 }, { "id": "2", "name": "Product 2", "price": 29.99 } ]

**3. Create a New Product**

**Endpoint:** **POST /api/products**

Create a new product by providing the product details in the request body.

**Request Body**

{ "id": "3", "name": "New Product", "price": 29.99 }

**Example**

POST http://localhost:3000/api/products

Content-Type: application/json

{ "id": "3", "name": "New Product", "price": 29.99 }

**Response**

{ "id": "3", "name": "New Product", "price": 29.99 }

**4. Update Product by ID**

**Endpoint:** **PUT /api/products/:id**

Update details of a specific product by providing the product ID and updated details in the request body.

Parameters

* **id** (string): The unique identifier of the product.

**Request Body**

{ "name": "Updated Product Name", "price": 39.99 }

**Example**

PUT http://localhost:3000/api/products/3

Content-Type: application/json

{ "name": "Updated Product Name", "price": 39.99 }

**Response**

{ "id": "3", "name": "Updated Product Name", "price": 39.99 }

**5. Delete Product by ID**

**Endpoint:** **DELETE /api/products/:id**

Delete a specific product by providing the product ID.

Parameters

* **id** (string): The unique identifier of the product.

**Example**

DELETE http://localhost:3000/api/products/3

**Response**

{ "id": "3", "name": "Updated Product Name", "price": 39.99 }

**Error Handling**

* If a requested product is not found, the API will respond with a **404 Not Found** status and an error message.
* If there are validation errors in the request data, the API will respond with a **400 Bad Request** status and an error message.
* Unexpected endpoints will result in a **404 Not Found** status and an error message.

**Response Format**

The API responds with JSON data for successful requests and error messages for unsuccessful requests.