**DOCUMENTATION FOR THE RETURN JOURNEY NODE.JS ASSIGNMENT**

* **INTRODUCTION**

The Node.js Products API is designed to facilitate the management of product-related data in a flexible and scalable manner. This API allows you to perform various operations, such as retrieving a list of products, adding new products, updating existing product details, and deleting products.

* **INSTALLATION**

Steps needed to install this project locally:-

1. Clone this Repository
2. Change the directory to the project folder.
3. Install dependencies with ‘npm install’ command.
4. Run the project with “npm run” command.

* **REST APIs**

**Endpoints:**

1. GET /api/products: Get a list of products.
2. GET /api/products/:id: Get details of a specific product.
3. POST /api/products: Create a new product.
4. PUT /api/products/:id: Update details of a specific product.
5. DELETE /api/products/:id: Delete a product.

**Authentication:**

As of this point, no authentication mechanism is implemented. No authentication mechanism was needed as well.

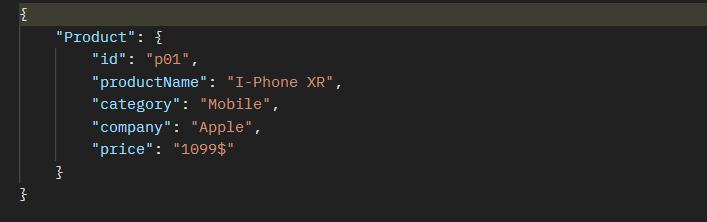
**REQUEST AND RESPONSE**

All the responses are in JSON format.

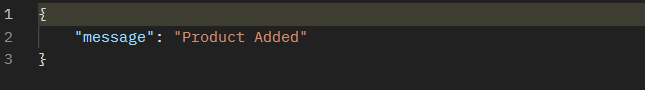
1. GET /api/products (STATUS: 200)



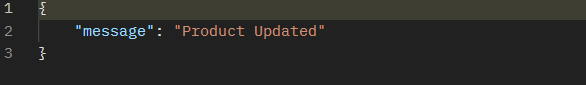
1. GET /api/products/:id (STATUS: 200)



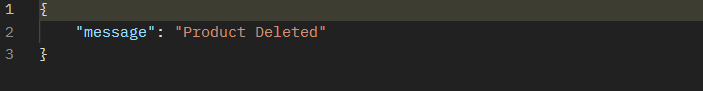
1. POST /api/products (STATUS: 201)



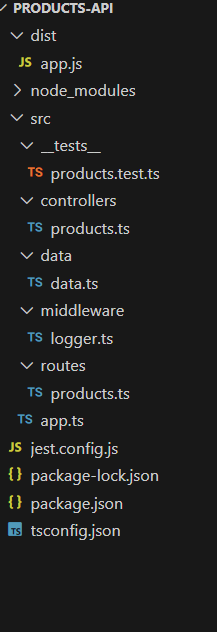
1. PUT /api/products/:id (STATUS: 201)



1. DELETE /api/products/:id (STATUS: 200)



* **FOLDER STRUCTURE**

****

* **DEPENDENCIES**

1. Express: Web framework for Node.js.
2. Morgan: HTTP request logger middleware for node.js.
3. Short-unique-id: Short unique id generator.

* **TESTING**

The libraries used for testing are: Jest and Supertest.

The Endpoints which are tested:

1. GET /api/products
2. GET /api/products/:id
3. POST /api/products
4. PUT /api/products/:id
5. DELETE /api/products/:id

All the test cases were passed.