Api Documentation

This is an api documentation of an ecommerce application where users can sign up, sign out and perform some basic tasks of an e-commerce application like fetch product details,manage cart and Place order or cancel it any time.

Introduction:

Our E-Commerce API enables you to build applications that interact with our online store. You can use the API to retrieve product information, place orders, manage user accounts, and more.

This application in Using docker as a development environment and for database please check the code accordingly.

The database schema file is provided to you in the github repo on github

Authentication:

To access data user must have to authenticate itself by either sign up or login.

To get authenticated and authorized user have to first sign up by providing name,email,password or (if already signed up then login by providing email and password).

After doing so user can access apis and get data.

End points

- POST /auth/register
    - Expects:{first_Name,last_Name,email,password}
    - Response:{email,name}

- POST /auth/login
    - Expects:{email,password},
    - Response:{email,name}
- GET /products
    - Response:(first 20 product)

      Note: user query parameters like limit,page

- GET /products/search?q=
    - Response: (products which matches the query parameter)
- GEt /products/categories
    - Response: (all categories list)
- GET /products/category/smartphones?limit=2
    - Response:(products from the category smartphone)

      Note: You can use query parameter like limit,page

- GET /products/{productId}

  Note: This is to get the full description of the product
    - Response: (full description of product with id)
- GET /carts
    - Response: (cart detail of user)
- POST /carts
    - Expects:{productId,quantity}
    - Response:{msg:acknowledge message}
- PUT /carts/{productId}

  Note: This is to update the quantity of product added to user’s cart
    - Expects:{quantity}

    - Response:{msg:acknowledge message}

- DELETE /carts/{productId}

  Note: This is to remove the product from the user's cart

    - Response:{msg:acknowledge message}
- GET /orders
    - Response: (get order list by user)

      Note: you can user query parameter like limit,page and sort=(asc/desc)
- POST /orders
    - Expects: {array of product details with quantity}
    - Response: {msg:operation acknowledge message}
- GET /orders/{orderId}
    - Response: {detail of user order whose order id is OrderId}
- PUT /orders/{orderId}

  Note: This is to cancel the order

    - Response: {msg:acknowledgement message}

Error Handling

Our API utilizes an error handler middleware to handle various errors and provide appropriate responses. Error responses include status codes, error messages.

Server Setup

- Install Dependencies: Run `npm i` or `npm install` to install required packages.
- Environment Variables: Configure the .env. file with necessary variables.
- Run the Server: Execute npm start to start the server.
- Server Listening: The server will be accessible at the specified `PORT`.

Conclusion

This documentation provides a basic overview of the available routes and functionality of out Node.js server. For detailed information on request and response formats,authentication process, and usage examples.