**PI Documentation for Your eCommerce Application**

**Base URL:** <http://localhost:2000/>

**Authentication**

* **JWT Token:** The API uses JSON Web Tokens (JWT) for authentication.
* **Login:** To obtain a JWT token, send a POST request to /login with the user's email and password.
* **Authorization Header:** Include the JWT token in the Authorization header of subsequent requests in the format Bearer <token>.

**Products Endpoints**

* **GET /products**
  + Retrieves a list of all products.
  + **Response:** Array of product objects with properties: id, title, price, description, category, image, rating.
* **GET /products/:id**
  + Retrieves a specific product by its ID.
  + **Response:** Product object with properties: id, title, price, description, category, image, rating.
* **POST /products**
  + Creates a new product. Requires authentication.
  + **Request Body:** Object with properties: title, price, description, category, image.
  + **Response:** Created product object with id property.
* **PUT /products/:id**
  + Updates an existing product by its ID. Requires authentication.
  + **Request Body:** Object with properties to update.
  + **Response:** Updated product object.
* **DELETE /products/:id**
  + Deletes a product by its ID. Requires authentication.
  + **Response:** Success message.

**Users Endpoints**

* **POST users/signup**
  + Registers a new user.
  + **Request Body:** Object with properties: name, email, password, gender, img.
  + **Response:** Success message.
* **POST users/login**
  + Logs in a user and returns a JWT token.
  + **Request Body:** Object with properties: email, password.
  + **Response:** Object with a token property.
* **GET /users/** 
  + Retrieves a list of all authenticated users, Requires authentication.
  + **Response:** Array of User object with properties: id, name, email, role, gender, img.
* **GET /users/:id**
  + Retrieves the authenticated user's information. Requires authentication.
  + **Response:** User object with properties: id, name, email, role, gender, img.
* **PUT /users/:id**
  + Updates the authenticated user's information. Requires authentication.
  + **Request Body:** Object with properties to update.
  + **Response:** Updated user object.
* **DELETE /users/:id**
  + Deletes the authenticated user's account. Requires authentication.
  + **Response:** Success message.

**Error Handling**

* The API returns appropriate HTTP status codes and error messages for different scenarios.
* Common error codes:
  + 400: Bad Request (invalid or missing data)
  + 401: Unauthorized (missing or invalid authentication token)
  + 403: Forbidden (unauthorized access)
  + 404: Not Found (resource not found)
  + 500: Internal Server Error (server-side error)