# API Documentation: User and Product Management

Warning: This document was partly created using chatgpt and may contain errors

This API provides a comprehensive system for managing users and products, integrating Firebase for data storage and handling. Below is a detailed documentation outlining authentication methods, data models (placeholders for customization), and available endpoints.

Inhoud

[API Documentation: User and Product Management 1](#_Toc155889920)

[Authentication 2](#_Toc155889921)

[Data Models 3](#_Toc155889922)

[User Endpoints 4](#_Toc155889923)

[Product Endpoints 5](#_Toc155889924)

[Notes 6](#_Toc155889925)

# Authentication

* **Method:** Token-based authentication
* **Usage:** Include the token in the request header as follows:
* Authorization: <token>
* **Acquiring Token:** Tokens are provided upon successful login via the /users/login endpoint. In the following format:
* token: <token>

# Data Models

1. **User Model**
   * id: Unique identifier
   * first\_name: First name of the user
   * last\_name: Last name of the user
   * email: Email address
   * phone\_number: Contact number
   * username: Unique username
   * password: Hashed password
   * token: Auth token
   * updated: Datetime
   * created: Datetime
2. **Product Model**
   * id: Unique identifier
   * name: Product name
   * description: Description of the product
   * expiration\_date: Expiry date
   * city: City
   * address: Address
   * user: Reference to the user who created the product (excluding sensitive data)
   * updated: Datetime
   * created: Datetime

*Deprectated properties which will later be deleted as they are included in the user field.*

* + first\_name: First name of the user
  + last\_name: Last name of the user
  + email: Email address
  + phone\_number: Contact number

# User Endpoints

1. **Create User**
   * POST /users
   * Description: Registers a new user
   * Request Body: Contains user details (first\_name, last\_name, email, phone\_number, username, password)
2. **User Login**
   * POST /users/login
   * Description: Authenticates a user and returns a token
   * Request Body: username and password
3. **User Logout**
   * POST /users/logout
   * Description: Logs out a user by invalidating the token
   * Requires Authentication: Yes
4. **Get User by ID**
   * GET /users/<id>
   * Description: Retrieves a user's details by ID
   * Requires Authentication: Yes
5. **Update User**
   * PUT /users/<id>
   * Description: Updates user details
   * Request Body: Updated user details
   * Requires Authentication: Yes
6. **Delete User**
   * DELETE /users/<id>
   * Description: Deletes a user's account
   * Requires Authentication: Yes

# Product Endpoints

1. **Get All Products**
   * GET /products
   * Description: Retrieves a list of all products
2. **Create Product**
   * POST /products
   * Description: Adds a new product
   * Request Body: Product details
   * Requires Authentication: Yes
3. **Get Product by ID**
   * GET /products/<id>
   * Description: Retrieves a single product by ID
4. **Delete Product**
   * DELETE /products/<id>
   * Description: Deletes a product
   * Requires Authentication: Yes

# Notes

* All endpoints returning data will provide it in JSON format.
* Error handling is implemented for scenarios like user/product not found, unauthorized access, and duplicate usernames.
* Passwords are stored securely using hashing.
* The current data models are placeholders and should be adjusted to fit the actual application schema.