User API Design Documentation

1. Introduction

The User API will handle basic end user operations (Create, Read, Update, Delete).

An end user will have limited CRUD operations available involving their own account such as being able to reset their password, update their personal details (add/delete addresses, etc), and delete their account.

1. Requirements

CRUD API to manage users stored in tbl\_users database, an end user will only be able to perform actions on their own account so each action will need to be associated with a valid token attached to their account.

Logging in and token management will be provisioned in a separate service, but the token store will be accessible to this user API for validation purposes.

All input/output data will be in JSON format, example below:

{

“id”: 10002,

“username”: “vfwood”,

“password\_hash”: “e443d94662sjems9”

“date\_added”: “2023-04-24 00:20:34”

“date\_last\_modified”: “2023-04-24 00:20:34”

“failed\_login\_attempts”: 0

“is\_deleted”: false

}

The user API will need to be secure; no plaintext passwords, using a valid password encoder which utilises an additional ‘salt’ to mitigate brute force cracking.

1. User API Endpoints

|  |  |  |  |
| --- | --- | --- | --- |
| METHOD | ENDPOINT | INPUT | OUTPUT |
| POST | /users | {  “username”: “user@email.com”,  “passwordHash”: “e443d94662sjems9”  “failedLoginAttempts”: 0  “dateAdded”: “2023-04-24 00:20:34”  “dateLastModified”: “2023-04-24 00:20:34”  “isDeleted”: false  } | **Success:**  HTTP 200 – OK  “Account Created Successfully”  **Error:**  HTTP 400 – Bad Request  Message dependent on reason for error. |
| GET | /users | na | {  {  “username”: “vfwood”,  “passwordHash”: “e443d94662sjems9”  “failedLoginAttempts”: 0  “dateAdded”: “2023-04-24 00:20:34”  “dateLastModified”: “2023-04-24 00:20:34”  “isDeleted”: false  },  {  “username”: “amyK123”,  “passwordHash”: “e44rrr662sjems9”  “failedLoginAttempts”: 1  “dateAdded”: “2023-04-24 00:20:34”  “dateLastModified”: “2023-04-24 00:20:34”  “isDeleted”: false  }  } |
| GET | /users/{id} | Integer: 10002 | {  “username”: “vfwood”,  “passwordHash”: “e443d94662sjems9”  “failedLoginAttempts”: 0  “dateAdded”: “2023-04-24 00:20:34”  “dateLastModified”: “2023-04-24 00:20:34”  “isDeleted”: false  } |
| PUT | /users/{id} | {  “username”: “vfwood”,  “passwordHash”: “e443d94662sjems9”  “failedLoginAttempts”: 0  “dateAdded”: “2023-04-24 00:20:34”  “dateLastModified”: “2023-04-24 00:20:34”  “isDeleted”: false  } | **Success:**  HTTP 204 – No Content  “Updated account details successfully”  **Error:**  HTTP 400 – Bad Request  Message dependent on reason for error. |

Users Table

* POST /users: Create a new user
* GET /users: Retrieve all users
* GET /users/{id}: Retrieve an existing user
* PUT /users/{id}: Update an existing user
* DELETE /users/{id}: Delete an existing user