User API Design Documentation

1. Introduction

The login API will offer a token to the calling client. The client will pass in a username and password hash the Login API will then verify that the credentials are indeed correct and then return a valid token to be passed around with requests to the other services.

1. Requirements

Login API will have one endpoint that responds to a POST method. The payload includes username and password hash and the request param includes an id for the user.

Firstly, upon successful verification of username and password hash the Login service will query the tbl\_user\_token table to locate a row with user\_id, if it finds a match and there is a valid token it will return the token (side note - any failed login attempts will be reset to 0).

On the first login attempt a token may not be created at this point it will insert into the tbl\_user\_token table to generate a new token. Within the tbl\_user\_token table we have an expiry\_date if that date is in the past the token is no longer valid and the system will generate a new token and update the tbl\_user\_token table with a new token and a new expiry\_date.

If you send through an incorrect password the service will increment a failed\_login\_attempts in the tbl\_user table and return an error message.

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

Input: **login/1**

{

“username”: “will123”,

“passwordHash”:” 5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8”

}

Output:

{

“success”: true,

“token”: “2a88364c-8108-4ea1-ae46-d90bfe060a7c”

“message”: “Login successful, token returned.”

}

1. User API Endpoints

| Method | Endpoint | Input | Output |
| --- | --- | --- | --- |
| POST | /login/{id} | {  “username”: ”will123”,  “password\_hash”: “5e884898da2804715 1d0e56f8dc6292773603d0d6aabbdd62a  11ef721d1542d8”  } | Success 200 OK:  {  “success”: true,  “token”: “2a88364c-8108-4ea1-ae46-d90bfe060a7c”  “message”: “Login successful, token returned.”  }  Error 401 Unauthorised (Incorrect Credentials):  {  “success”: false,  “token”: null,  “message”: “Error logged in with incorrect credentials.”  }  Error 403 Forbidden (Login attempt limit exceeded)  {  “success”: false,  “token”: null,  “message”: “Error reached maximum login attempts please contact our support team via email”  }  Error 400 Bad Request  {  “success”: false,  “token”: null,  “message”: “Error, reached maximum login attempts please contact our support team via email”  }  Error 404 Not Found  {  “success”: false,  “token”: null,  “message”: “Error user does not exists”  } |

Users Table

* **POST** */login/{id}* verifies username and password and returns valid token