

## Fluency Box API Documentation

To use the Fluency Box Rest API, your application will make HTTP requests to restful end points. The request and response payload content type is in most cases *application/json*.

The Rest API tries to adhere as closely as possible to standard HTTP and REST conventions in its use of HTTP verbs.

<b>GET</b>	Used to retrieve a resource
<b>POST</b>	Used to create a new resource
<b>PUT</b>	Used to update an existing resource
<b>DELETE</b>	Used to delete an existing resource

### Error Handling

All responses will contain a json key called **"status"**. The value of this key will either be **"success"** or **"fail"**, the latter status showing a failure. If a request fails, another key called **"message"** will describe the reason for the failure. Appropriate status codes will be returned:

<b>400</b>	Bad Request
<b>401</b>	Unauthorized
<b>404</b>	Resource Not Found
<b>500</b>	Internal Server Error

## 1. Login Request

**Request Type: POST**

**Request Path: /login**

The Login request requires client to authenticate using HTTP basic auth.

Pass the Email Address & Password to receive a valid response.

### Success Response - 200

```
{
  "refresh_token":
  "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJVSUQiOiIzMGZiYmVkMi01MzU2LTQ1NmItOTZiNy03N2Q0MDAwMmMyMjMiLCJ0b2t1b190eXB1IjoieF9yZWZyZXNoX3Rva2VuIiwiaXhwIjoxNTY1NjQ2MjQzfQ.-3Z8Zrnj-bGMI6iM4KkKPCl0zHo4iXjrcUCzgGA-AgU",
  "status": "success",
  "token":
  "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJVSUQiOiIzMGZiYmVkMi01MzU2LTQ1NmItOTZiNy03N2Q0MDAwMmMyMjMiLCJ0b2t1b190eXB1IjoieF9hY2Nlc3NfdG9rZW4iLCJleHAiOiE1NjU1NTk5MDN9.RnjJZum07MombYd9bjUwhgCacZDdj0ZyzHPEmXJPzt8",
  "uid": "30fbbed2-5356-456b-96b7-77d40002c223"
}
```

The above response returns a JSON body which contains the refresh\_token, status, token & uid will be returned.

### Failure Response – 401

Invalid Credentials

The header in this case will contain the below Key and Values

Key	Value
WWW-Authenticate	Basic realm="Login required!"

For every subsequent request, ensure the **token** from the above response is provided in the header.

Key	Value
x-access-token	Valid token

## 2. Refresh Token

**Request Type:** POST

**Request Path:** /refresh\_token

The access token has a short expiry, the refresh token has a longer expiry and this has to be stored on the client front end and be used to request for a new access token when it expires.

To obtain a new access token, pass the below values in the header of the request. This request takes no JSON parameters.

A successful response will return a new refresh and access token which should be used for the subsequent requests. Please note that the new refresh token should be stored and used when requesting for next access token.

### Request

Key	Value
x-refresh-token	Refresh token value

### Success Response – 200

```
{
  "refresh_token":
  "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJVSUQ0IiZMGZiYmVkMi01MzU2LTQ1NmItOTZiNy03N2Q0MDAwMmMyMjMiLCJ0b2t1b190eXB1IjoieF9yZWZyZXNoX3Rva2VuIiwiaXhwIjoxNTY1NjQ2NDkzfQ.h5dpL8v1wNtjp-CcYoYImDAmkHbXR72pRG-CVMRWcSY",
  "status": "success",
  "token":
  "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJVSUQ0IiZMGZiYmVkMi01MzU2LTQ1NmItOTZiNy03N2Q0MDAwMmMyMjMiLCJ0b2t1b190eXB1IjoieF9yZWZyZXNoX3Rva2VuIiwiaXhwIjoxNTY1NjQ2NDkzfQ.h5dpL8v1wNtjp-CcYoYImDAmkHbXR72pRG-CVMRWcSY",
  "uid": "30fbbbed2-5356-456b-96b7-77d40002c223"
}
```

### Failure Response – 401

```
{
  "message": "Token is invalid",
  "status": "fail"
}
```

### 3. Get A Specific User

**Request Type:** GET

**Request Path:** /users/:id

Use the uid returned in the login request to obtain details about the user.

Request

http://127.0.0.1:5000/users/30fbbed2-5356-456b-96b7-77d40002c223

Success Response – 200

```
{
  "status": "success",
  "user": {
    "email_address": "sanjum308@hotmail.com",
    "first_name": "Sanjay",
    "last_name": "Mehta",
    "phone_number": "+254 7228189991",
    "profile_picture": "default_image.jpg",
    "uid": "30fbbed2-5356-456b-96b7-77d40002c223",
    "user_name": "Sanjay7"
  }
}
```

Failure Response – 404

```
{
  "message": "No user found",
  "status": "fail"
}
```

#### 4. Create A User (Registration)

**Request Type:** POST

**Request Path:** /users

**Content Type:** application/json

##### Request

```
{
  "first_name" : "Anthony",
  "last_name" : "M",
  "email_address" : "Anthony@hotmail.com",
  "phone_number" : "+254 7228189991",
  "user_name" : "Anthony",
  "password" : "12345"
}
```

##### Success Response - 201

```
{
  "message": "New User Created",
  "status": "success"
}
```

##### Failure Response – 400

In case an email has already been registered, the below error will be returned

```
{
  "message": "Email already registered",
  "status": "fail"
}
```

In case a username has already been registered, the below error will be returned

```
{
  "message": "User name already registered",
  "status": "fail"
}
```

In case the JSON request body does not have a particular data value, for example “email\_address”, an email\_address missing error will occur. Same errors will occur if any of the other fields are missing in the request.

```
{
  "message": "No email address in request",
  "status": "fail"
}
```

## 5. Update A User

**Request Type:** PUT

**Request Path:** /users/:id

**Content Type:** application/json

Request

http://127.0.0.1:5000/users/9c9f4d3c-62b7-4cab-b06b-14a82bd025b1

```
{
  "first_name" : "Hiro",
  "last_name" : "Mendo",
  "email_address" : "hiromendo@gmail.com",
  "phone_number" : "+254 7228189991",
  "user_name" : "Hiro7"
}
```

Success Response – 200

```
{
  "message": "User Updated Successfully",
  "status": "success"
}
```

Failure Response – 404

```
{
  "message": "No user found",
  "status": "fail"
}
```

Failure Response – 400

In case the JSON request body does not have a particular data value, for example “email\_address”, an email\_address missing error will occur. Same errors will occur if any of the other fields are missing in the request.

```
{
  "message": "No email address in request",
  "status": "fail"
}
```

## 6. Update Password

**Request Type:** PUT

**Request Path:** /users/:id/password

**Content Type:** application/json

### Request

http://127.0.0.1:5000/users/30fbbed2-5356-456b-96b7-77d40002c223/password

```
{  
  "password" : "12345"  
}
```

### Success Response – 200

```
{  
  "message": "Password Updated Successfully",  
  "status": "success"  
}
```

### Failure Response – 400

```
{  
  "message": "No password in request",  
  "status": "fail"  
}
```

### Failure Response – 404

```
{  
  "message": "No user found",  
  "status": "fail"  
}
```

## 7. Update Profile Picture

**Request Type:** PUT

**Request Path:** /users/:id/profile\_picture

**Content Type:** application/x-www-form-urlencoded

This request requires a file – the profile picture of the user.

The key name for this file is **profile\_picture\_file**

Request

```
http://127.0.0.1:5000/users/8265e368-71d7-406e-9b56-fe5a36c7e6f7/profile_picture
```

Success Response – 200

```
{
  "message": "successfully updated",
  "status": "success"
}
```

Failure Response – 400

```
{
  "message": "No file found",
  "status": "fail"
}
```

Failure Response – 404

```
{
  "message": "No user found",
  "status": "fail"
}
```



## 8. Delete User

**Request Type:** DELETE

**Request Path:** /users/:id

Request

```
http://127.0.0.1:5000/users/a2f82c41-4ecd-4524-8f26-6a666103ef99
```

Success Response – 200

```
{  
  "message": "User Deleted Successfully",  
  "status": "success"  
}
```

Failure Response – 404

```
{  
  "message": "No user found",  
  "status": "fail"  
}
```

## 9. Reset Password Request – No token required

**Request Type:** POST

**Request Path:** /reset\_password

**Content Type:** application/json

Request

http://127.0.0.1:5000/reset\_password

```
{
  "email" : "sanjum308@hotmail.com"
}
```

Success Response – 200

```
{
  "message": "An email has been sent with instructions to reset your password.",
  "status": "success"
}
```

Failure Response – 400

If no email key is found in the JSON request body, the below error will be thrown

```
{
  "message": "No email in request",
  "status": "fail"
}
```

Failure Response – 404

```
{
  "message": "No user found",
  "status": "fail"
}
```

## 10. Update New Password – No token required

**Request Type:** PUT

**Request Path:** /users/:email/password\_reset

**Content Type:** application/json

Request

http://127.0.0.1:5000/users/sanjum308@hotmail.com/password\_reset

```
{  
  "password" : "12345"  
}
```

Success Response – 200

```
{  
  "message": "Password Updated Successfully",  
  "status": "success"  
}
```

Failure Response – 400

```
{  
  "message": "No password in request",  
  "status": "fail"  
}
```

Failure Response – 404

```
{  
  "message": "No user found",  
  "status": "fail"  
}
```