

# FLASK TEST

## API Technical Documentation

Version History	Date	Change Reason
Version 1.0	18/06/2024	First version

## Table of Contents

<b>Introduction</b>	1
1. Base URLs	1
2. Auth Endpoint	1
3. User Endpoint	1
4. Token's Validity Period	1
<b>API Documentation</b>	2
1. Register New User	2
2. Authentication	3
3. List All Users	5
4. Error Handling	6

# Introduction

Flask Test API is a RESTful web API for interacting with front-end users. All the data exchanged between front-end's user and the API is JSON over HTTP.

## 1. Base URLs

The base URLs for the API is: <http://localhost:5000>

All the requests on this URL use either POST or GET method and we need to add the specific end points to this URL. Example, if you want to register new user, you will have to make POST call to the below API: <http://localhost:5000/auth/register> with the parameters given in the section specific to this call.

## 2. Auth Endpoint

Following are the end points of authentication

- /auth/register
- /auth/login

## 3. User Endpoint

Following are the end points of user after login

- /user/all
- /user/all?page=<number>&per\_page<number>

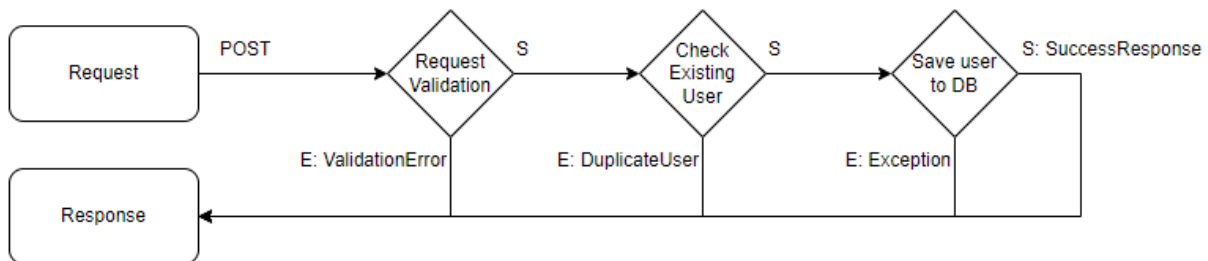
## 4. Token's Validity Period

The current validity period of token is **60** minutes

# API Documentation

The following sections discuss request and response parameters for all individual APIs

## 1. Register New User



- Endpoint: **/auth/register**
- Method: **POST**

### Request

Parameter Name	Data Type	Remark
username	String	Min=5
email	String	Email format validation
password	String	Min=5; Max=20

Example:

```
{
  "username": "testuser1",
  "email": "testuser1@example.com",
  "password": "password"
}
```

### Response

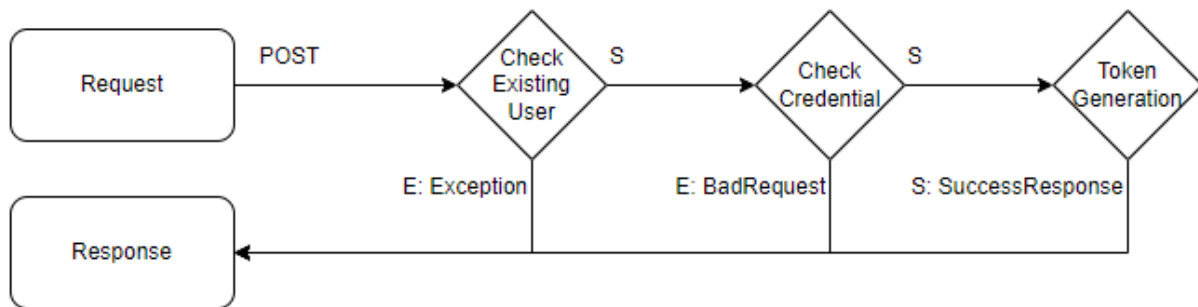
Parameter Name	Data Type	Remark
status	String	SUCCESS, ERROR
message	Array	Nullable

data	Object	
------	--------	--

Example:

```
{
  "data": {
    "email": "testuser1@example.com",
    "username": "testuser1"
  },
  "message": [
    "User has been successfully registered"
  ],
  "status": "SUCCESS"
}
```

## 2. Authentication



- Endpoint: **/auth/login**
- Method: **POST**

### Request

Parameter Name	Data Type	Remark
username	String	Min=5
password	String	Min=5; Max=20

Example:

```
{
  "username": "testuser",
  "password": "password"
}
```

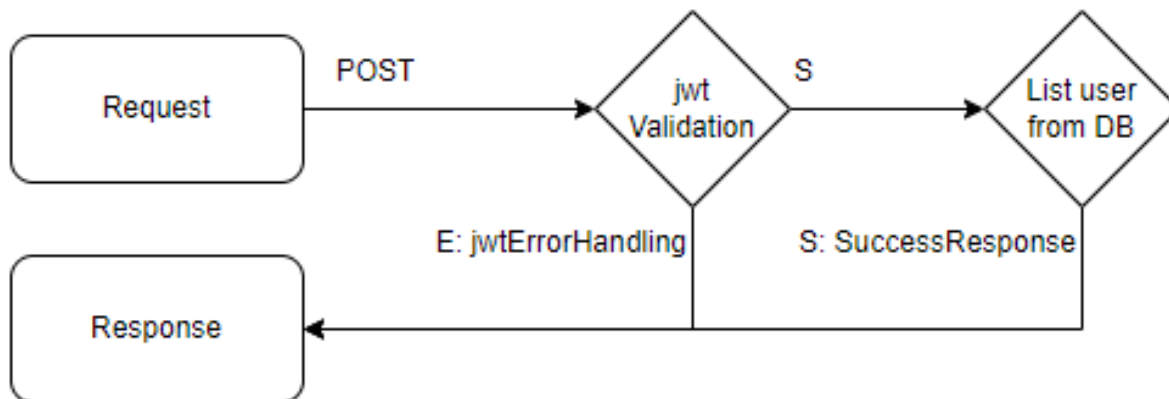
## Response

Parameter Name	Data Type	Remark
status	String	SUCCESS, ERROR
message	Array	Nullable
data	Object	

Example:

```
{
  "data": {
    "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmFsc2UsIm1hdCI6MTcxODY5ODgwNSwianRpIjoianNk4OGVhYzItZmNlNy00NmM3LTg1NmEtZmM0ODZmOTY1MTM3IiwidHlwZSI6ImFjY2VzcyIsInN1YiI6ImRlM2NjYTQ1LTE0YTctNGFiZC1iZjJmLTZhMGZiMzFiMTY5MyIsIm5iZiI6MTcxODY5ODgwNSwiY3NyZiI6ImQyYzZlYmMyLWU1YTItNDBlNi1iNGJmLTNmYmM0ZmYxNmNlMCIsImV4cCI6MTcxODcwMjQwNX0.zN5kIYPreOZWwv
hG2ALnQJK-8jgK56UsIInpFdqFCQQ",
    "email": "testuser@example.com",
    "expires_in": "Tue, 18 Jun 2024 11:20:05 GMT",
    "username": "testuser"
  },
  "message": [
    "Logged in successfully"
  ],
  "status": "SUCCESS"
}
```

### 3. List All Users



- Endpoint:  
**/user/all**  
**/user/all?page=<number>&per\_page<number>**

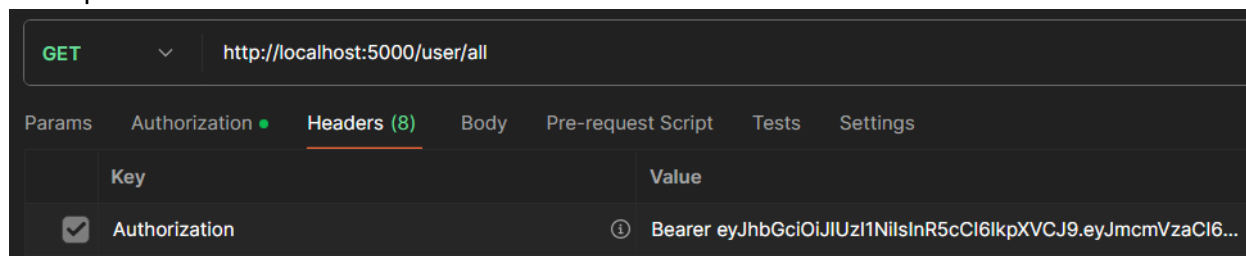
Where <number> is Integer.

- Method: **POST**

#### Request Headers

Key	Value	Remark
Authentication	Bearer <token>	<b>token</b> was generated after successful login in endpoint “/auth/login”

Example:



#### Response

Parameter Name	Data Type	Remark
----------------	-----------	--------

status	String	SUCCESS, ERROR
message	Array	Nullable
data	Object	

Example:

```
{
  "data": {
    "total": 1,
    "user_data": [
      {
        "email": "testuser@example.com",
        "id": "de3cca45-14a7-4abd-bf2f-5a0fb31b1693",
        "username": "testuser"
      }
    ]
  },
  "message": null,
  "status": "SUCCESS"
}
```

## 4. Error Handling

### Response

Parameter Name	Data Type	Remark
status	String	SUCCESS, ERROR
type	String	
message	String	

Example:

```
{
  "message": "Missing Authorization Header",
  "status": "ERROR",
  "type": "missing_token"
}
```