# Auth

## Login API (username and password based)

Authenticate users and generate JWT token.

### ### Request

- \*\*Method\*\*: POST

- \*\*URL\*\*: `/login`

- \*\*Content Type\*\*: application/json

### #### Request Body

```json

{

"identity": "string",

"password": "string"

}

### Responses

#### Success Response (200 OK)

{

"success": true,

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV\_adQssw5c"

}

#### Error Responses

##### 400 Bad Request: If the request body is missing or invalid.

{

"error": "Bad Request",

"message": "Invalid request body"

}

##### 401 Unauthorized: If the credentials are incorrect.

{

"error": "Unauthorized",

"message": "Invalid credentials"

}

##### 500 Internal Server Error: If there is a server-side issue.

{

"error": "Internal Server Error",

"message": "An unexpected error occurred"

}

## Register

Register new users.

### ### Request Header

- \*\*Method\*\*: POST

- \*\*URL\*\*: `/register`

- \*\*Content Type\*\*: application/json

### #### Request Body

```json

{

"firstName": "string",

"lastName": "string",

"username": "string",

"password": "string",

"phoneNo": "string",

"professions": "[{

id: String,

moreDetails: Object

}]",

"email": "string",

"gender": "string" (M/F)

}

### Responses

#### Success Response (201 Created)

{

"error": false,

"message": "User registered successfully",

“user” : {},

“token”: string

}

#### Error Responses

##### 400 Bad Request: If the request body is missing or invalid.

{

"error": "Bad Request",

"message": "request body should not be empty "

}

##### 400 Bad Request: If email, username and phoneNo are not provided

{

"error": "Bad Request",

"message": "No username, email and phoneNo are provided. You should provide either email or phoneNo or username"

}

##### 400 Bad Request: If email or phoneNo are not provided in valid format

{

“error”: “Bad Request”,

“message”: “User validation failed: phoneNo: Please fill a valid phone number”

}

##### 409 Conflict: If the username is already in use.

{

"error": "Conflict",

"message": "username is already registered ",

“errorField”: “username”

}

##### 409 Conflict: If the phoneNo is already in use.

{

"error": "Conflict",

"message": "PhoneNo is already registered",

“errorField”: “phoneNo”

}

##### 409 Conflict: If the email is already in use.

{

"error": "Conflict",

"message": "email is already registered”,

“errorField”: “email”

}

##### 500 Internal Server Error: If there is a server-side issue.

{

"error": "Internal Server Error",

"message": "An unexpected error occurred"

}

## Available Professions API

### ### Request

Method: GET

URL: /api/profession/all

Content Type: application/json

### ### Response

"professions":

[

{

"\_id": String,

"professionTitle": String,

"industry": String ",

"professionDescription": String,

"svgUrl": "https://cloud-object-storage-cos-standard-rhm.s3.jp-tok.cloud-object-storage.appdomain.cloud/1715373970310-carpenter.svg"

}

]

# Job Board

## Post a job

### Request Header

Method: POST

URL: /job

Content-Type: multipart/form-data

### Request Body

JSON

{

title: 'String',

description: 'String',

location: 'String',

salaryRange: {

min: 'Number',

max: 'Number'

},

employmentType: 'String', // ["full-time", "part-time", "contract"]

deadlineDate: 'Date',

requirements: "[String, String, ….]",

benefits: '[String, String,.,.,.]'

}