

# Authentication API Routes

---

1. [Register](#)
2. [Login](#)
3. [Refresh](#)
4. [Logout](#)
5. [Forgot Password](#)
6. [Reset Password](#)
7. [Verify Email](#)
8. [Is Verified](#)

## Register

**Status:** Operative **Description:** Registers a user in case they use credentials. It creates the user in db and returns the access and refresh tokens needed for authentication. It also sends the link to email provided to verify it. **Endpoint:** `/auth/register` **Method:** `POST` **Requires authentication:** `false`

- **Body:**

```
{
  "email": "user@example.com",
  "name": "User Example",
  "password": "password123",
  "confirmPassword": "password123",
  "provider": "credentials"
}
```

- **Response: 200 OK:**

```
{
  "success": true,
  "user": {
    "_id": MongoDB ObjectId,
    "email": string,
    "name": string,
    "profilePicture": string | null,
    "isVerified": boolean,
    "createdAt": Date,
    "updatedAt": Date
  },
  "access": {
    "token": "eyJhbGciOiJSUzI1NiIsInR...",
    "expires": Date
  },
  "refresh": {
    "token": "eyJhbGciOiJSUzI1NiIsInR...",
    "expires": Date
  }
}
```

```
}  
  
}
```

**400 Bad Request:** Invalid input or missing fields.

```
{  
  "success": false,  
  "error": "Invalid input"  
}
```

**409 Conflict:** Email already in use.

```
{  
  "success": false,  
  "error": "Email already in use"  
}
```

**409 Conflict:** Passwords don't match or email is already in use.

```
{  
  "success": false,  
  "error": "Passwords must be the same"  
}
```

**500 Internal Server Error:** Internal server error.

```
{  
  "success": false,  
  "error": "when registering the user: ${e}"  
}
```

---

## Login

**Status:** Operative **Description:** Authenticates the user with provided credentials and returns JWT tokens for session management. There are two types of login in this platform, the credentials and the Google token one.

**Endpoint:** `/auth/login` **Method:** `POST` **Requires authentication:** `false`

- **Body:**

```
// For the credentials login
{
  "email": "user@example.com",
  "password": "password123",
  "provider": "credentials",
}

// For the login with Google
// We create user in db in case they don't exist - ONLY IN GOOGLE LOGIN
{
  "email": "user@example.com",
  "name": "Some Name",
  "picture": "url.com" | null,
  "provider": "google",
  "token": "eyJhbGciOiJSUzI1NiIsInR..." // Google token provided by authJS in
the front
}
```

- **Response: 200 OK:**

```
{
  "success": true,
  "user": {
    "_id": MongoDB ObjectID,
    "email": string,
    "name": string,
    "profilePicture": string | null,
    "password": null,
    "authProvider": "google" | "credentials",
    "createdAt": Date,
    "updatedAt": Date
  },
  "access": {
    "token": "eyJhbGciOiJSUzI1NiIsInR...",
    "expires": Date
  },
  "refresh": {
    "token": "eyJhbGciOiJSUzI1NiIsInR...",
    "expires": Date
  }
}
```

**400 Bad Request:** Invalid input or missing fields.

```
{
  "success": false,
```

```
{
  "error": "Email or password missing"
}
```

**404 Not Found:** Email not found (only in credentials login).

```
{
  "success": false,
  "error": "User not found"
}
```

**500 Internal Server Error:** Internal server error.

```
{
  "success": false,
  "error": "Something went wrong when logging the user with [provider]:
  ${e}"
}
```

---

## Refresh

**Status:** Operative **Description:** Receives a refresh token, verifies it is not expired and generates new access tokens. This system is for added security in the requests. **Endpoint:** `/auth/refresh` **Method:** `POST` **Requires authentication:** `true` // refresh token

- **Body:**

```
{
  "refreshToken": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXV..."
}
```

- **Response: 200 OK:**

```
{
  "success": true,
  "access": {
    "token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXV...",
    "expires": Date
  },
  "refresh": {
    "token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXV...",
    "expires": Date
  }
}
```

```
}
```

**400 Bad Request:** Refresh token not provided

```
{
  "success": false,
  "error": "Refresh token not provided"
}
```

**401 Unauthorized:** Refresh token expired

```
{
  "success": false,
  "error": "Refresh token"
}
```

**500 Internal Server Error:** Internal server error.

```
{
  "success": false
}
```

---

## Forgot Password

**Status:** Operative **Description:** This endpoint is in charge of sending a link the user's email in case they forgot their password, so they can reset it. Only available if the provider is "credentials". Send the wanted email in the body. **Endpoint:** `/auth/forgot-password` **Method:** POST **Requires authentication:** false

- **Body:**

```
{
  "email": "example@mail.com"
}
```

- **Response: 200 OK:**

```
{
  "success": true,
  "data": "eyJhbGciOiJSUzI1Ni..."
}
```

```
// should send an email too with the link to reset the password
}
```

**400 Bad Request:** Email not provided

```
{
  "success": false,
  "error": "Email was not provided"
}
```

**404 Not Found:** Email not registered in db.

```
{
  "success": false,
  "error": "User not found. Email is not registered in the platform"
}
```

**500 Internal Server Error:** Internal server error.

```
{
  "success": false,
  "error": "Something went wrong when resetting the password ${e}"
}
```

---

## Reset Password

**Status:** Operative **Description:** Using the token sent with forgot-password flow, this endpoint verifies that token is still valid and then changes the password to the new one the user setted. **Endpoint:** `/auth/reset-password` **Method:** `POST` **Requires authentication:** `false`

- **Body:**

```
{
  "email": "example@mail.com",
  "password": "1234",
  "confirmPassword": "1234",
  "resetPasswordToken": "eyJhbGciOiJIUzI1NiIsInR..."
}
```

- **Response: 200 OK:**

```
{
  "success": true,
  "data": "eyJhbGciOiJSUzI1NiIu..."
  // should send an email too with the link to reset the password
}
```

**400 Bad Request:** Some information from the body not provided

```
{
  "success": false,
  "error": "All information was not provided"
}
```

**401 Unauthorized:** Rest password token is not valid

```
{
  "success": false,
  "error": "Reset password token is not valid"
}
```

**404 Not Found:** Email not registered in db.

```
{
  "success": false,
  "error": "User not found"
}
```

**409 Conflict:** Passwords must be the same

```
{
  "success": false,
  "error": "Passwords must be the same"
}
```

**500 Internal Server Error:** Internal server error.

```
{
  "success": false,
  "error": "Something went wrong when resetting the password ${e}"
}
```

## Verify Email

**Status:** Operative **Description:** This endpoint verifies the email of a user based on a token provided **Endpoint:** `/auth/verify-email` **Method:** POST **Requires authentication:** true // verifyEmailToken necessary

- **Body:**

```
{
  "email": "example@mail.com",
  "verifyEmailToken": "eyJhbGciOiJIUzI1NiIsInR..."
}
```

- **Response: 200 OK:**

```
{
  "success": true,
}
```

**400 Bad Request:** Some information from the body not provided

```
{
  "success": false,
  "error": "Email or token not provided"
}
```

**401 Unauthorized:** Verify email token not valid

```
{
  "success": false,
  "error": "Verify email token is not valid"
}
```

**404 Not Found:** Email not registered in db.

```
{
  "success": false,
  "error": "User not found"
}
```

**500 Internal Server Error:** Internal server error.



```
{
  "success": false,
  "error": "Something went wrong when verifying the email: ${e}"
}
```

---

## Is Verified

**Status:** Operative **Description:** This endpoint verifies the email of a user based on a token provided **Endpoint:** `/auth/is-verified` **Method:** GET **Requires authentication:** true // accessToken

- **Response: 200 OK:**

```
{
  "success": true,
}
```

**401 Unauthorized:** Verify email token not valid

```
{
  "success": false,
  "error": "Verify email token is not valid"
}
```

**404 Not Found:** Email not registered in db.

```
{
  "success": false,
  "error": "User not found"
}
```

**500 Internal Server Error:** Internal server error.

```
{
  "success": false,
  "error": "Something went wrong when verifying the email: ${e}"
}
```