API Documentation

Authentication

This document provides the technical specification for the API endpoints related to user authentication, session management, and authorization as defined in *auth.py*.

1. **User Login**
   Authenticates a user and provides session tokens
   - **Endpoint:** POST /auth/login
   - **Description:** Validates a user's username and password. If the credentials are correct, it generates and returns a JWT access token and a refresh token.
   - **Authentication:** Not required.

   Request Body

Field	Type	Required	Description
username	String	Yes	The user's unique username.
password	String	Yes	The user's plain-text password

   Example Request:

```
{
    "username": "a_valid_user",
    "password": "their_correct_password"
}
```

   Responses

   - **200 OK (Success)**
     - **Description:** Returned when the credentials are valid.
     - **Body:**

```
{
  "access_token": "..."
  "refresh_token": "..."
```

```
}
```

- **400 Bad Request (Invalid Input)**
  - **Description:** Returned if required fields are missing or the input is malformed.
  - **Body:**

```
{
  "Message": "Invalid input".
  "errors": {
      [
          "Missing data for required field.
      ]
   }
}
```

- **401 Unauthorized (Invalid Credentials)**
  - **Description:** Returned if the username does not exist or the password is incorrect.
  - **Body:**

```
{
  "message": "Invalid credentials"
}
```

2. **Refresh Access Token**

   Issues a new access token using a valid refresh token.
   - **Endpoint:** POST /auth/refresh
   - **Description:** Allows a client to obtain a new, short-lived access token without requiring the user to log in again.
   - **Authentication:** Refresh Token Require. The request must include a valid refresh token in the *Authorization* header.

   Request Body

   (None)

   Responses

- **200 OK (Success)**
  - **Description:** Returned when the refresh token is valid.
  - **Body:**

```
{
  "access_token": "..."
}
```

- **401 Unauthorized (Invalid Token)**
  - **Description:** Returned if the refresh token is missing, invalid, or expired.
  - **Body:**

```
{
  "msg": "Token has expired"
}
```

3. **User Logout**

   Invalidates the current user's access token.

   - **Endpoint:** DELETE /auth/logout
   - **Description:** Logs the user out by adding their current access token's unique identifier (***jti***) to a server-side blocklist. This prevents the token from being used for any further authenticated requests.
   - **Authentication: Access Token Required.** The request must include a valid access token in the *Authorization* header.

   Request Body

   (None)

   Responses

   - **200 OK (Success)**
     - **Description:** Returned when the token is successfully added to the blocklist.
     - **Body:**

```
{
  "message": "Successfully logged out!"
}
```

- **401 Unauthorized (Invalid Token)**
  - **Description:** Returned if the access token is missing, invalid or expired.
  - **Body**

```
{
  "msg": "Missing Authorization Header"
}
```