### Authentication ****API Documentation****

#### ****1. Custom Login Endpoint****

* **Endpoint**: POST /auth/api/login/
* **Description**: This endpoint allows users to log in by providing their username and password. If the credentials are correct, a token is returned for authenticated access to other parts of the system.

**Request**

* **Method**: POST
* **Body**:
  + username: (string) The username of the user.
  + password: (string) The password of the user.

**Request Example**

{

"username": "example\_user",

"password": "password123"

}

**Response**

* **Status 200**: When the user is authenticated successfully, the response will include the token.

{

"token": "abcd1234efgh5678ijkl9012mnop3456"

}

* **Status 401**: If the credentials are incorrect, the response will include an error message.

{

"error": "Invalid credentials"

}

**Logical Steps**:

1. The user sends a POST request with the username and password.
2. The system authenticates the credentials using Django’s authenticate function.
3. If authentication is successful:
   * A token is created or retrieved using Token.objects.get\_or\_create(user=user).
   * The token is returned in the response.
4. If authentication fails:
   * A 401 Unauthorized status is returned with the error message "Invalid credentials".

#### ****2. Logout Endpoint****

* **Endpoint**: POST /api/logout/
* **Description**: This endpoint allows authenticated users to log out by deleting their authentication token, invalidating their session.

**Request**

* **Method**: POST
* **Headers**:
  + Authorization: (string) The token obtained from logging in.

**Request Example**

Authorization: Token abcd1234efgh5678ijkl9012mnop3456

**Response**

* **Status 200**: Successfully logged out.

{

"message": "Successfully logged out"

}

**Logical Steps**:

1. The user sends a POST request with an Authorization header containing the token.
2. The system checks the validity of the token.
3. The token is deleted using request.user.auth\_token.delete().
4. The response contains a success message confirming the user is logged out.

### ****Swagger Documentation****

swagger: '2.0'

info:

title: Authentication API

description: API for logging in and logging out of the application.

version: 1.0.0

paths:

/auth/api/login/:

post:

summary: Login using username and password.

description: Authenticates a user and returns a token.

consumes:

- application/json

parameters:

- in: body

name: body

description: Login credentials

required: true

schema:

type: object

properties:

username:

type: string

example: example\_user

password:

type: string

example: password123

responses:

200:

description: Successful login. Token returned.

schema:

type: object

properties:

token:

type: string

example: abcd1234efgh5678ijkl9012mnop3456

401:

description: Invalid credentials. Login failed.

schema:

type: object

properties:

error:

type: string

example: Invalid credentials

/api/logout/:

post:

summary: Log out the user by deleting the authentication token.

description: Invalidates the user session by deleting the token.

consumes:

- application/json

parameters:

- in: header

name: Authorization

description: The token obtained after login

required: true

type: string

example: Token abcd1234efgh5678ijkl9012mnop3456

responses:

200:

description: Successful logout. Token deleted.

schema:

type: object

properties:

message:

type: string

example: Successfully logged out

401:

description: Unauthorized if no valid token is provided.

schema:

type: object

properties:

error:

type: string

example: Authentication credentials were not provided.