User API Documentation

In this document, you will find every endpoints of the user acount managing API explained with parameters and URI

Quick Access

Register User

This route is used to register a new user in database, it implies to send user data. It also sends a mail to the new user to confirm his account

```
/register
```

Method : Post

*Params : *

data : A Json string requiring the following values:

```
nom : The last name
username : The username
email : The Email address
password : The password in clear
birthdate : The birthdate in the format YY-mm-dd
prenom : The first name
```

No mandatory encoding

Params exemple :

```
{"username":"username","email":"foo.bar@foobar.fr","password":"123","birthdate":
"1995-01-01","prenom":"Michel","nom":"Dupont"}
```

*Request Exemple : *

```
    http://localhost:8081/register?data=
{"username":"XxX_DarkmasterPGM72_XxX","email":"jphilippe.alle@free.fr","password
":"123","birthdate":"1995-08-13","prenom":"Michel","nom":"Dupont"}
```

Returns an error message if the account hasn't been created

Log user

This route is used to, with email and password, return user data and it's tokens

```
/login
```

Method : Post

*Params : *

email : the email address
password : the password

No mandatory encoding

Request Exemple

```
http://localhost:8081/login?email=jphilippe.alle@free.fr&pass=123
```

*Response Exemple : *

```
{"id":74,"username":"Jean","nom":"Dupont","prenom":"Michel","birthdate":"1995-
08-
13","email":"foo.bar@foobar.fr","password":"cG1Xa1dTQkNMNTFCZmtobjc5eFB1SOJLSHov
L0g2QittWTZHOS9laWV1TTO=","level":1,"has_conf":0,"adding_time":16551994978858300
00,"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjcOLCJpYXQiOjE2NTU
yODY3NTgsImV4cCI6MTY1NTMxNTU1OHO.Et9zTH88Uhk-eU6G6rsrmK-
JHtQB6grpSbAKDrIZfOM","refresh":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySW
QiOjcOLCJpYXQiOjE2NTUyODY3NTgsImV4cCI6MTY4Njg0NDM1OHO.B7X86QEyB_Sr_USJ9o4-
z7SDq3zTV-oH4kPn55gOKiO"}
```

Update userprofile

This root is used to update user profile data by sending userdata

TOKEN REQUIRED

```
/update
```

Method : Post

*Params *

data : A Json string requiring the following values:

```
nom : The last name
username : The username
email : The Email address
password : The password hashed n sha256 and encoded in base64
birthdate : The birthdate in the format YY-mm-dd
prenom : The first name
id : the database id of the user
```

id : The database if of the user (returned during login)

**Params Exemple : **

```
{"id":74,"username":"Jean","nom":"Dupont","prenom":"Michel","birthdate":"1995-
08-
13","email":"jphilippe.alle@free.fr","password":"CG1Xa1dTQkNMNTFCZmtobjc5eFB1SOJ
LSHovL0g2QittWTZHOS9lawV1TT0=","level":}
```

**Request Exemple : **

```
    http://localhost:8081/register?data=
{"username":"XxX_DarkmasterPGM72_XxX","email":"jphilippe.alle@free.fr","password
":"123","birthdate":"1995-08-13","prenom":"Michel","nom":"Dupont"}
```

**Response Exemple : **

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

Get permissions level of a user

This route is used to get the permission level of the user

TOKEN REQUIRED

```
    /get_level
```

Method : Post

**Params : **

id : The database id of the user

**Request Exemple : **

```
    http://localhost:8081/get_level?id=74
```

Response exemple

```
    {"success":true,"message":"ok","level":1,"id":74}
```

Change user elevation

This route is used to edit the permission level of the user

TOKEN REQUIRED

```
/change_user_elev
```

Method : Post

*Params : *

id : The id of the user in database
level : The new level to give to the user

*Request Exemple : *

```
http://localhost:8081/change_user_elev?id=74&level=1
```

*Response Exemple : *

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

Get user info

This toute is used to retrieve all datas of a user profile

TOKEN REQUIRED

```
/getuser
```

Method : Post

*Params : *

id : The user ID in the database

Request Exemple :

```
http://localhost:8081/getuser?id=74
```

Response Exemple :

```
{"id":74,"username":"Philippe","nom":"Dupont","prenom":"Michel","birthdate":"1995-08-13","email":"jphilippe.alle@free.fr","password":"cG1Xa1dTQkNMNTFCZmtobjc5eFB1SOJLSHovL0g2QittWTZHOS9laWV1TTO=","level":1,"has_conf":0,"adding_time":1655199497885830000}
```

Regenerate User Token

This route is used to return regenerated token for the user, requires refreshtoken

REFRESH TOKEN REQUIRED

```
/regen_token
```

Method : Post

*Params : *

id : The user ID in the database

*Request Exemple : *

```
http://localhost:8081/regen_token?id=74
```

Response Exemple :

```
{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjcOLCJpYXQiOjE2NTUyO
TAzMTYsImV4cCI6MTY1NTMxOTExNn0.B2ZYbqfcdxY3izQ9wStegLmHi2Pp-Sk_R6z65Xmwxbk"}
```

Delete user account

This route is used to delete user account

TOKEN REQUIRED

```
/delete
```

Method : Post

*Params : *

id : The user ID in the database

*Request Exemple : *

```
http://localhost:8081/delete?id=74
```

Response Exemple :

```
{"success":false,"message":"User not in database"}
```