

# API AREA

## Preface

This Project is made by Benjamin Piriac, Mark-Antoine Wittling, Matieu Dèru, Stànisslass Hérron, Jean Bougnarel and Mikael Gerard, for Epitech.

## Introduction

The AREA is a simple dashboard with services. Each services have action or reaction. Some of them will need OAuth2. It is build with docker. The purpose of the project is to creating AREA :

- **Action** : A predefined event triggering a reaction (like a new post from someone on reddit)
- **REAction** : Predefined event supposed to be trigger by a action (like post a new image on imgur)

The project is compose of :

- **Client** : call the api and display informations, run on port 8081
- **API / Server**: manage database and call externe api and send information to the client, running on port 8080
- **Mobile client** : available on localhost:8081/client.apk
- **Mong data base**: A data base by mongodb, available on the port 27017

## API Routes

The following commands explain there ous use

### Connect - GET

<http://server:8080/connectUser>

**Body :**

**username** : username of the user  
**password** : password of the user

Connect a user, return success or fail.

**return exemple :**

```
{  
  success : True  
  message : "User connected"  
}
```

**Status Success :** 201

**Status Error :** 401

## Create user - POST

<http://server:8080/createUser>

### Body :

**username** : username of the user  
**password** : password of the user

Connect a user, return success or fail.

### return exemple :

```
{  
  success : True  
  message : "User created"  
}
```

**Status Success** : 201

**Status Error** : 401

All next request need a specific header :

**access\_token** : Token recieve on the connection

## Create area - POST

<http://server:8080/CreateArea>

### Body :

**action** : Action for the area

**reaction** : Reaction for the area

All parameter needed by the action and the reaction, exemple :

**city** : City for the weather

**channel\_id** : Channel to post a message

**message** : message to post

Create a AREA.

### return exemple :

```
{  
  success : True  
  message : "Area created successfully"  
}
```

**Status Success** : 201

**Status Error** : 401

## Get area- GET

<http://server:8080/GetArea>

Get all AREA of the user.

return exemple :

```
[
  {
    _id: "5e25e05471f19f0107f86672",
    area_id: "405ope7bk5mpo3hi",
    user_id: "405ope1mk5mpevm0",
    action: "weather_time",
    action_params: {
      city: "Nantes",
      time: "10:10"
    },
    reaction: "discord_send_message",
    reaction_params: {
      channel_id: "12435",
      message: "the weather is {weather}"
    }
  }
]
```

Status Success : 201

Status Error : 401

## Update area - POST

<http://server:8080/UpdateArea>

Body :

**area\_id** : Id of the Area

**action** : Action for the area

**reaction** : Reaction for the area

All parameter needed by the action and the reaction, exemple :

**city** : City for the weather

**channel\_id**: Channel to post a message

**message**: message to post

Update a AREA.

return exemple :

```
{
  success: True
  message: "Area updated"
}
```

Status Success : 201

Status Error : 401

## Delete area - Delete

<http://server:8080/CreateArea>

### Body :

**area\_id** : Id of the area

Delete a AREA.

### return exemple :

```
{  
  success : True  
  message : "Area deleted"  
}
```

**Status Success :** 201

**Status Error :** 401

## Add token - POST

<http://server:8080/auth/:app>

### Param :

**app** : Name of the application the user is connected

### Body :

**access\_token** : Token returned by the auth2

**refresh\_token**: Token returned by the auth2

**expires\_in**: Expiration returned by the auth2

Save token from authentication in differents services.

### return exemple :

```
{  
  success : True  
  message : "Token saved"  
}
```

**Status Success :** 201

**Status Error :** 401

## Get services - GET

<http://server:8080/auth/getServices>

Get the services the user is connected to.

**return exemple :**

```
{  
  Discord : True,  
  Reddit : True,  
  Imgur : False,  
  Steam : True  
}
```

**Status Success :** 201

**Status Error :** 401