

# Project Octopus

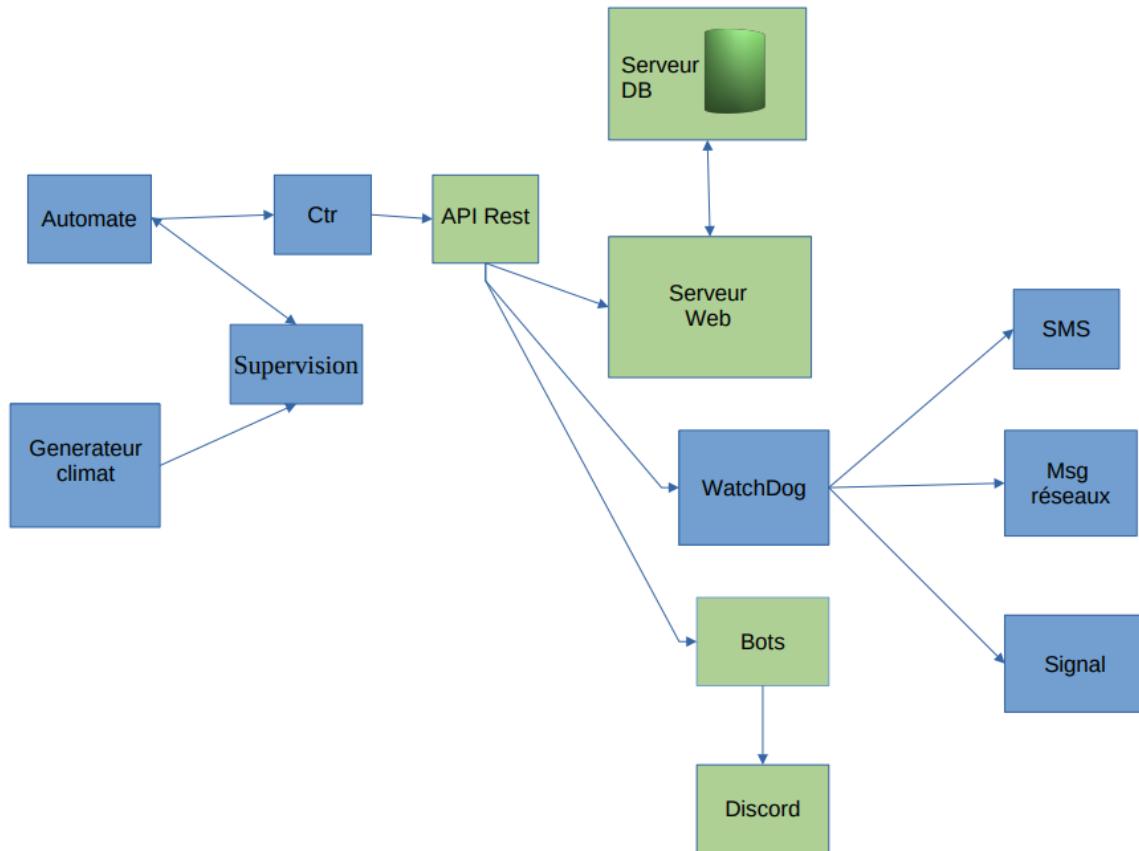
## Tables des matières

Contexte: .....	2
Architecture système .....	2
API REST .....	4
Principes de bases .....	4
Arborescence : .....	4
Comment Interroger les APIs des Ecolabs ? .....	6
Routage : .....	7
DB : Data Base .....	8
Structure de la Base de données : .....	8
Comment Communiquer avec la Base de Donnée ? .....	8
Web .....	8
En attente pour la Partie de Luca.....	8
Rendu final .....	9
Accueil .....	9
Détail .....	10
Experiences .....	12
Utilisateurs .....	13
Bot Octopus: .....	14
Arborescence .....	14
Les Commands : .....	14
Exemples : .....	15

# Contexte:

Le projet Octopus a été créé dans le but de faciliter la surveillance des expériences menées dans les cellules climatiques (ECOLAB). Il offre des fonctionnalités telles que la consultation à distance et un système d'alerte pour suivre le déroulement des expériences.

## Architecture système



**NOTE**

Voici un schéma illustrant la structure de notre projet. Les blocs en vert représentent les éléments intégrés à notre projet Octopus, tandis que les blocs en bleu ont déjà été réalisés par Mr. chollet

### Automate

Système de pilotage et d'aquisition des actionneurs et capteurs des chambres climatiques. L'automate est autonome : il régule les chambres climatiques selon des consignes.

## Generateur climat

Système de généralisation de climats qui permet de générer des fichiers contenant les variations de paramètre de climat : température hygrométrie concentration de CO<sub>2</sub>, etc...

Le générateur de climat transforme des données météo provenant de différentes stations dans le format supporté par les chambres climatiques.

## API Rest

Le serveur API REST permet au serveur web de récupérer des paramètres de température en temps réel au format JSON

Serveur DB



Le serveur de base de données permet d'avoir une sauvegarde des expériences en cours et terminées dans les cellules, tout en gérant les rôles des utilisateurs.

Serveur Web

Le serveur web est une interface permettant aux clients de consulter les écolabs à distance. Ils peuvent vérifier les températures actuelles des cellules, ainsi que leurs expériences. Les administrateurs ont la possibilité de modifier ou d'ajouter des expériences à distance.

## WatchDog

Système de surveillance qui permet de prévenir le personnel en cas de défaillance sur le déroulement d'une expérience.

Le watchdog surveille les paramètres climatiques et permet d'envoyer des messages sous

forme de SMS.

## Bots

Les bots ont un rôle de communication avec les clients, échangeant des informations. Les utilisateurs peuvent demander au bot la température actuelle d'une cellule ou solliciter le lien de notre site web. Dans le cadre de ce projet, nous avons codé un bot Discord.

## API REST

Une API REST repose sur des principes fondamentaux incluant l'architecture orientée ressources avec des identifiants URI, la représentation des données en JSON ou XML, la communication via les méthodes HTTP standard (GET, POST, PUT, DELETE), le principe de communication sans état, et une interface uniforme définissant des conventions pour les URI, les méthodes HTTP et les représentations des ressources.

### Principes de bases

- Architecture Orientée Ressources : Identification des éléments par des URI.
- Représentation des Données : Utilisation de formats standard (JSON, XML) pour représenter les ressources.
- Méthodes HTTP Standard : Utilisation des méthodes GET, POST, PUT, DELETE pour interagir avec les ressources.
- Communication Sans État : Chaque requête contient toutes les informations nécessaires, et le serveur ne garde pas d'état entre les requêtes.
- Interface Uniforme : Définition de conventions pour les URI, les méthodes HTTP et les représentations des ressources.

#### NOTE

Dans ce projet, nous avons utilisé uniquement les méthodes HTTP **GET** en format **JSON**

### Arborescence :

```
└── .idea  
    ├── encodings.xml  
    ├── misc.xml  
    └── modules.xml  
    └── Octopus.iml  
    └── workspace.xml
```

```
    |   Commons
    |   |   __init__.py
    |   |
    |   |   Config
    |   |   |   config.py
    |   |   |   config_json.py
    |   |   |   __init__.py
    |   |
    |   |   __pycache__
    |   |   |   __init__.cpython-38.pyc
    |
    |   config
    |   |   climate_settings.json
    |   |   errors.json
    |
    |   |   ECOLAB_1 ... (tout les fichier json qui contient les parametre adapter
pour interroger l'automate )
    |   |   |   E1C1.json
    |   |   |   E1C2.json
    |   |   |   E1C3.json
    |   |   |   E1TH.json
    |   |   |   ECOLAB_1.json
    |
    |   |   ECOLAB_2 ... (tout les fichier json qui contient les parametre adapter
pour interroger l'automate )
    |   |   |   E2C1.json
    |   |   |   E2C2.json
    |   |   |   E2C3.json
    |   |   |   E2TH.json
    |   |   |   ECOLAB_2.json
    |
    |   |   ECOLAB_3 ... (tout les fichier json qui contient les parametre adapter
pour interroger l'automate )
    |   |   |   E3C1.json
    |   |   |   E3C2.json
    |   |   |   E3C3.json
    |   |   |   E3TH.json
    |   |   |   ECOLAB_3.json
    |
    |   |   ECOLAB_4 ... (tout les fichier json qui contient les parametre adapter
pour interroger l'automate )
    |   |   |   E4C1.json
    |   |   |   E4C2.json
    |   |   |   E4C3.json
    |   |   |   E4TH.json
    |   |   |   ECOLAB_4.json
    |
    |   |   ECOLAB_5 ... (tout les fichier json qui contient les parametre adapter
pour interroger l'automate )
    |   |   |   E5C1.json
    |   |   |   E5C2.json
```

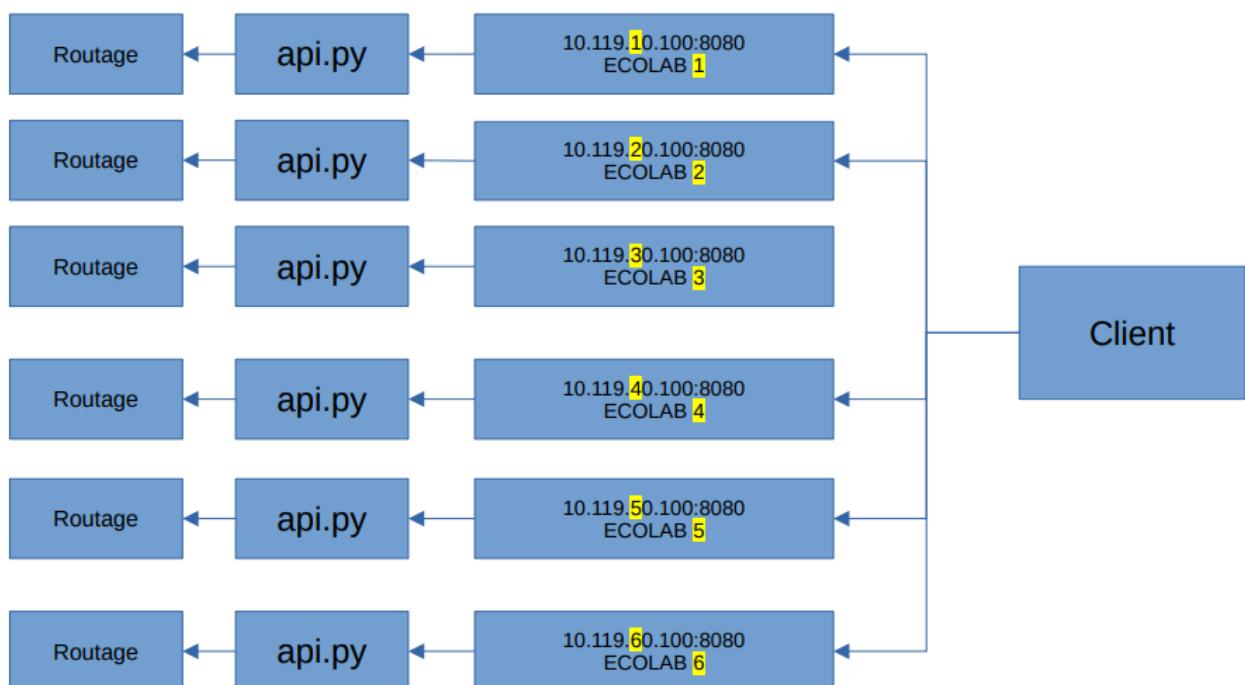
```

|   |   E5C3.json
|   |   E5TH.json
|   |   ECOLAB_5.json
|   |
|   |   ECOLAB_6 ... (tout les fichier json qui contient les parametre adapter
|   |   pour interroger l'automate )
|   |       E6C1.json
|   |       E6C2.json
|   |       E6C3.json
|   |       E6TH.json
|   |       ECOLAB_6.json
|   |
|   |   api.py ... (Programme Python qui récupère et retourne les API)
|   |   base_error.py
|   |   cell.py ... (Class Cell)
|   |   climate_settings.py
|   |   config.json
|   |   E1C1.json
|   |   ecolab.py
|   |   plc.py
|   |   thermo.py

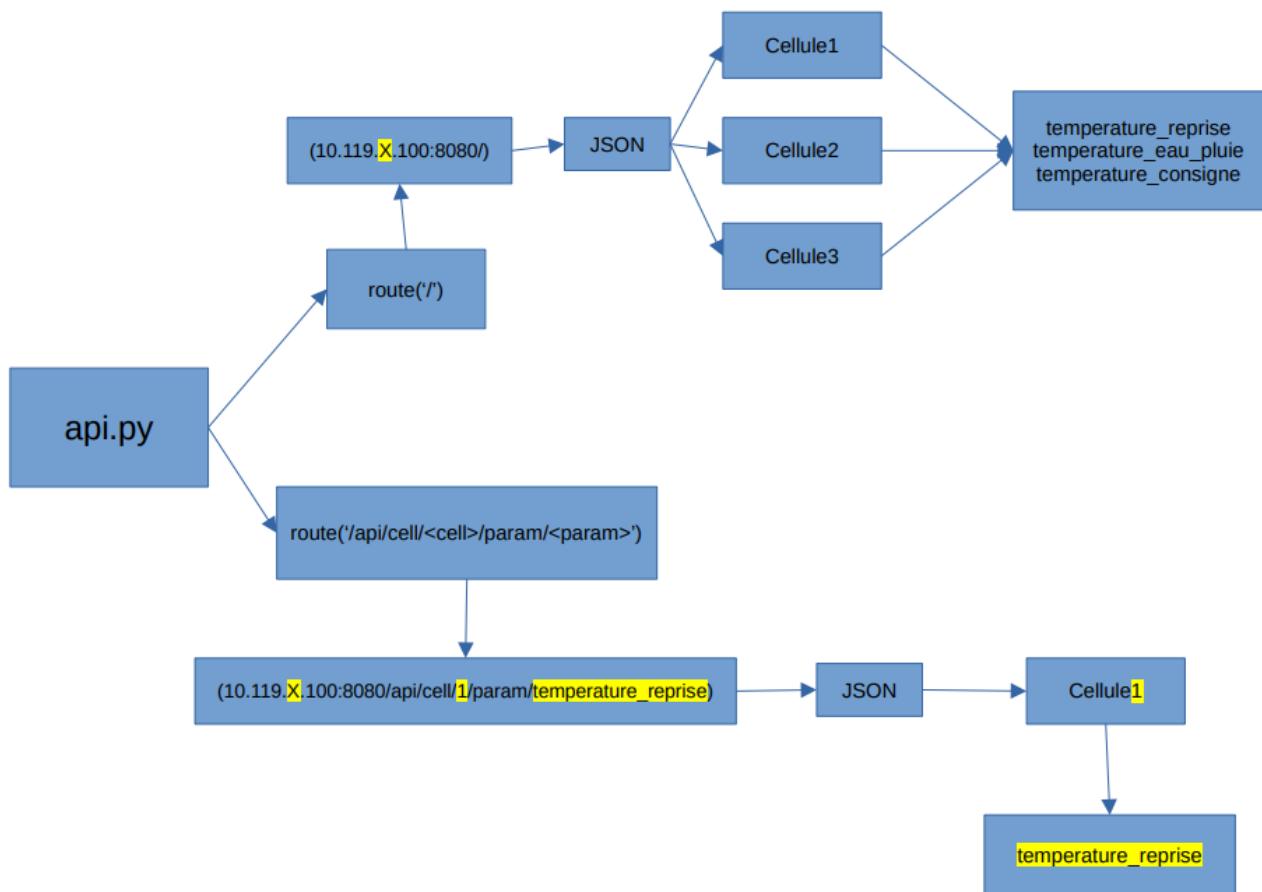
```

## Comment Interroger les APIs des Ecolabs ?

Chaque écolab est identifié par une adresse IP. Lorsqu'on interroge l'adresse IP de l'écolab, elle renvoie les paramètres des cellules qu'elle contient.



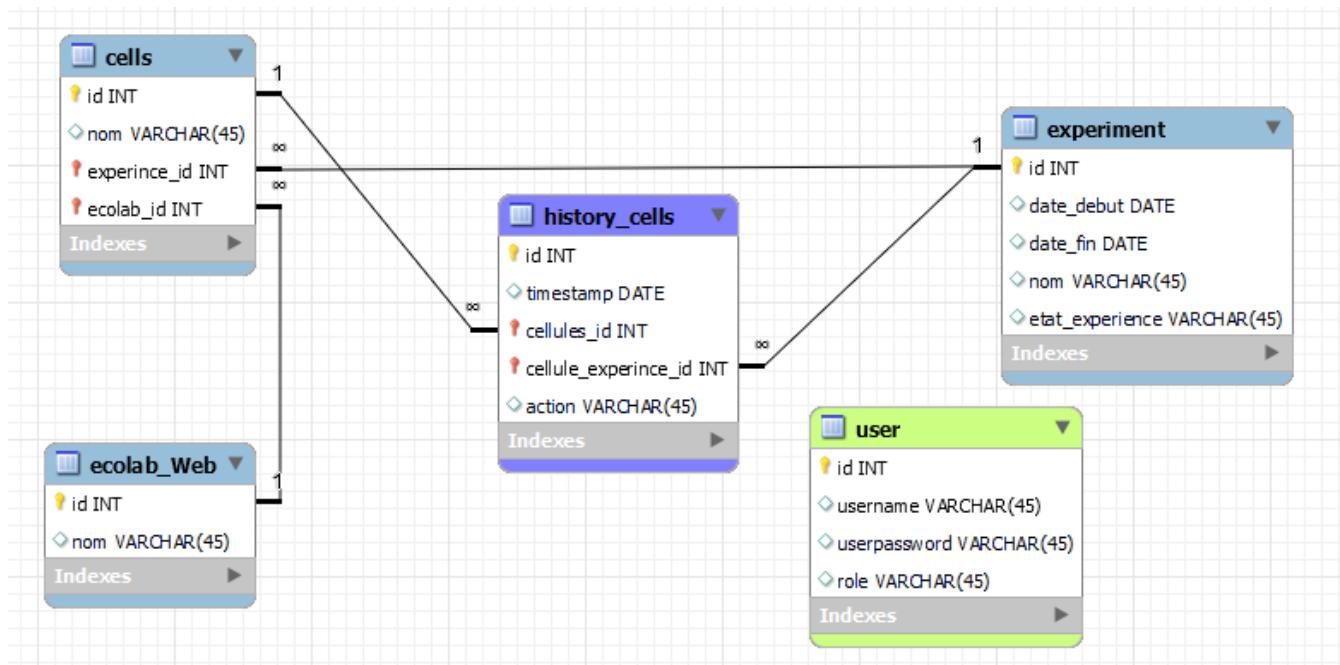
## Routage :



Le fichier `api.py` contient deux routages. Le premier routage à la racine renvoie tous les paramètres de toutes les cellules de l'écolab. Le deuxième routage retourne uniquement le paramètre souhaité d'une seule cellule.

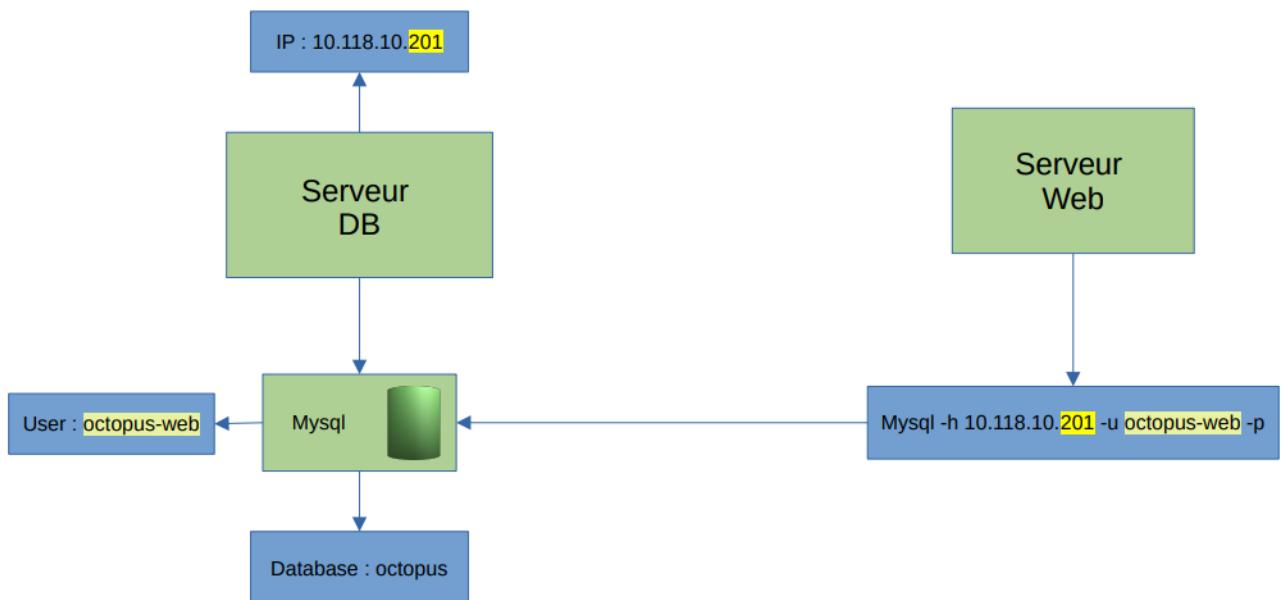
# DB : Data Base

Structure de la Base de données :



Comment Communiquer avec la Base de Donnée ?

Lorsque vous souhaitez communiquer avec la base de données du serveur DB, il suffit d'ajouter l'adresse IP du serveur DB devant l'utilisateur.



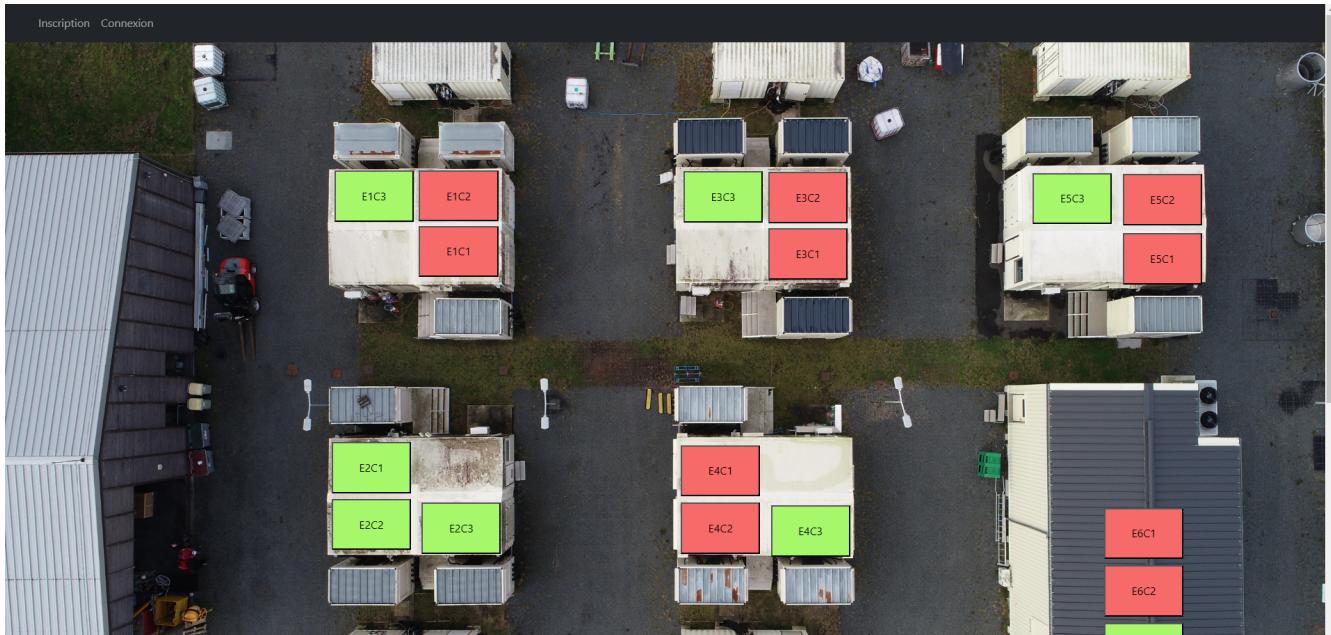
# Web

En attente pour la Partie de Luca.....

# Rendu final

## Accueil

accessible par des visiteurs et des administrateurs.



### Nous suivre

- Facebook
- Twitter
- Instagram
- YouTube

[Accèsibilité](#) [Crédits](#) [Gestion des cookies](#) [Données personnelles](#) [Mentions légales](#)

Page d'inscription :

Nom d'utilisateur

Mot de passe

S'inscrire

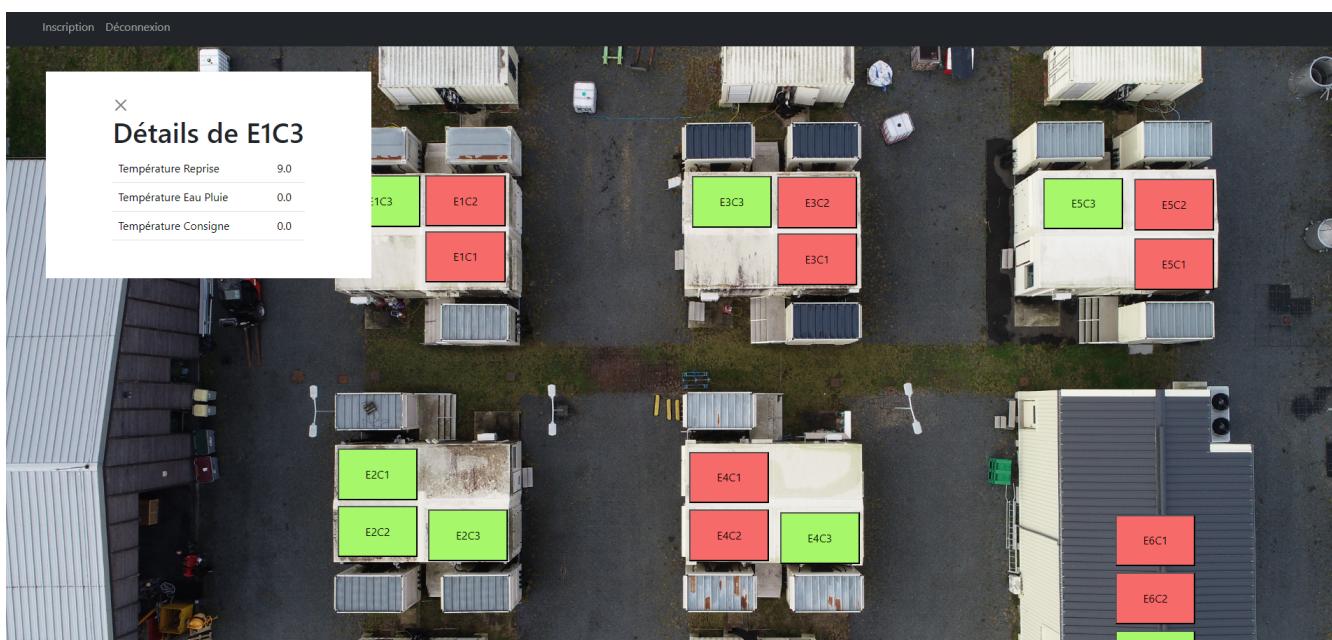
*Page de connexion :*

Nom d'utilisateur

Mot de passe

Se connecter

## Détail



uniquement pour l'administrateur :



[Retour](#)

### Détail de E2C3

Temperature Reprise	16.1
Temperature Eau Pluie	10.9
Temperature Consigne	0.0

### L'historique

Date et Heure	Les expériences	Date de Debut	Date de Fin	Status	Action
2024-01-24 11:42:38	Sylvie	2023-10-01	None	Terminé	None
2024-02-06 10:22:32	EMBL-EXPOCR	2023-12-13	2024-01-24	En cours	Ajout d'une nouvelle expérience à la cellule

Affecter une nouvelle expérience :

EMBL-EXPOCR ()

[Confirmer](#)

*La liste des expériences pour les affecter à la cellule :*

[Retour](#)

## Détail de E2C3

Temperature Reprise	16.1
Temperature Eau Pluie	10.9
Temperature Consigne	0.0
Choisir l'expérience	
EMBL-EXPOCER (à venir)	
Gente-POP (à venir)	
MangWarm (à venir)	
B2 (En cours)	
Mines-Pluie (En cours)	
Peace (à venir)	
Lysimètre (à venir)	
Bacs à évaporation (à venir)	
Sybio (à venir)	
Optisol (à venir)	
Microbe For (à venir)	
Climate (à venir)	
AgroServ (à venir)	
Oasis (à venir)	
Sylvie (à venir)	
Azerty (En cours)	
ml (à venir)	
Ip (à venir)	
nj (à venir)	
EMBL-EXPOCER (à venir)	

[Confirmer](#)

## Experiences

[Retour](#)[Ajouter une expérience](#)

### Les Expériences

Id	Nom	Date de Début	Date de Fin	Etat de l'expérience	Action
1	EMBL-EXPOCER	2023-12-13	2024-01-24	à venir	<a href="#">Modifier</a>
2	Gente-POP	2023-08-01	None	à venir	<a href="#">Modifier</a>
3	MangWarm	2023-09-01	None	à venir	<a href="#">Modifier</a>
4	B2	2024-01-01	None	En cours	<a href="#">Modifier</a>
5	Mines-Pluie	2023-11-01	None	En cours	<a href="#">Modifier</a>
6	Peace	2024-02-01	None	à venir	<a href="#">Modifier</a>
7	Lysimètre	None	None	à venir	<a href="#">Modifier</a>
8	Bacs à évaporation	None	None	à venir	<a href="#">Modifier</a>
9	Sybio	None	None	à venir	<a href="#">Modifier</a>
10	Optisol	None	None	à venir	<a href="#">Modifier</a>
11	Microbe For	None	None	à venir	<a href="#">Modifier</a>
12	Climate	None	None	à venir	<a href="#">Modifier</a>

*Modifier une experience:*

Nom  
EMBL-EXPOCR

Date de Debut  
13/12/2023

Date de Fin  
24/01/2024

Etat  
à venir

[Confirmer](#) [Retour](#)

*Crée une experience:*

### Ajouter une nouvelle Experience

Nom  
Entrée le nom de l'experience

Date de Debut  
jj/mm/aaaa

Date de Fin  
jj/mm/aaaa

Etat  
à venir

[Confirmer](#)

## Utilisateurs

[Retour](#)

### Les Utilisateurs

Nom	Role	Action
admin	admin	<a href="#">Modifier</a>
cmoi	utilisateur	<a href="#">Modifier</a>
laa	visiteur	<a href="#">Modifier</a>

*Modifier le rôle ou le nom d'un utilisateur:*

Nom  
admin

Nouveau Role  
Admin

Confirmer Retour

## Bot Octopus:



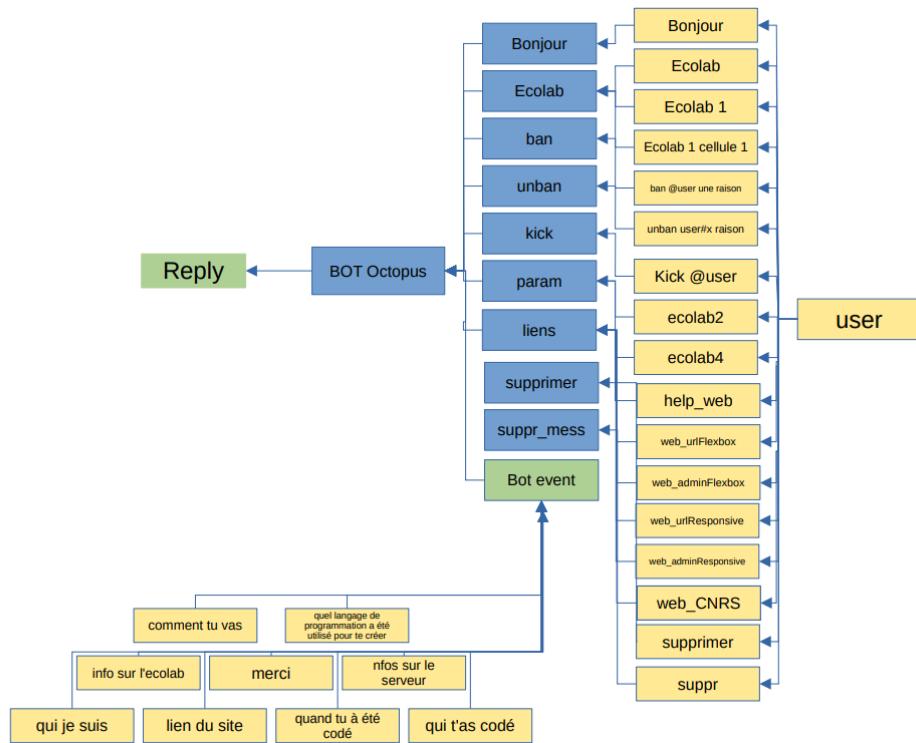
Le bot Octopus est présent dans le salon API du serveur Octopus sur Discord.

## Arborescence

```
| api.py  
| bot.py  
| key.py  
| Liens.json  
  
└── __pycache__  
    └── api.cpython-311.pyc  
    └── key.cpython-311.pyc
```

## Les Commands :

Voici la liste des commandes qui vous permettent de communiquer avec le bot :



## Exemples :

