



→ Pour savoir comment exécuter le serveur, il faut vous référer à la procédure d'installation Serveur/Client.

→ Pour la documentation technique DocString, celle-ci est directement située dans le code serveur (fichier « serveur.py »).

I – Démarrage du serveur :

Lorsque l'on démarre le serveur, nous avons cet affichage :

```
Connexion à la BDD réussie.  
Serveur lancé sur l'hôte 0.0.0.0, port 24793.  
  
Email Administrateur:
```

Ceci est une demande d'authentification administrateur pour avoir accès à la boucle de commandes serveur.

Pour s'authentifier, nous utiliserons le compte initialisé dans la procédure d'installation Serveur/client :

Email : admin@admin.com
MDP : admin

Une fois l'authentification passée, voici l'affichage renvoyé :

```
Connexion à la BDD réussie.  
Serveur lancé sur l'hôte 0.0.0.0, port 24793.  
  
Email Administrateur: admin@admin.com  
Mot de passe: admin  
Authentification réussie.  
  
[ADMIN] Entrez une commande :
```

Nous sommes désormais dans la boucle de gestion du serveur, où peuvent être effectuées des commandes d'administration diverses.

II – Commandes administrateur :

Le serveur permet d'effectuer, via des commandes, les actions suivantes :

- Bannir un client via son email et toutes les adresses IP qui lui sont liées.
- Révoquer le bannissement d'un client via son email.

- Kick un client pour une durée définie via son email et les adresses IP liées.
- Révoquer le kick d'un client via son email.

- Autoriser l'accès à un salon à un client via le nom du salon et son email.
- Révoquer l'accès à un salon à un client via le nom du salon et son email.

- Arrêt de toutes les connexions clients et fermeture du serveur.

Effet des commandes :

- Bannissement : déconnecte immédiatement le client du serveur et lui interdit toute authentification et inscription définitivement, jusqu'à révocation.

- Kick : déconnecte immédiatement le client du serveur et lui interdit toute authentification et inscription temporairement (durée en minutes) OU jusqu'à révocation.

- Grant : autorise le client à accéder à un salon spécifique, ou lui en révoque l'accès.

- Kill : coupe toutes les connexions clients (en les prévenant) et ferme le serveur au bout de 5 secondes.

Fonctionnement technique des commandes :

- Bannissement et Kick : Les sanctions sont enregistrées dans une table dédiée de la base de données (sanctions) et collaborent avec une autre table dédiée à l'historique des adresses IP connues du client (historique_ip). Ces tables sont parcourues, aux bons endroits, afin de vérifier si le client et ses adresses IP connues sont sous l'effet d'une sanction ou non, pour le contrôle d'accès.

L'expiration du Kick est calculée automatiquement par le programme qui fait le delta entre l'horodatage du moment où la sanction « Kick » est infligée, et l'horodatage du moment où le client essaie de s'authentifier/s'inscrire, en comparant à la durée spécifiée initialement.

- Grant : une table de la base de données est dédiée aux accès des salons publics (membres_salons_publics). Lors de l'utilisation de la commande, le programme ajoute ou révoque l'accès du client au salon spécifié, et se charge automatiquement de vérifier ladite table pour les logiques de contrôle d'accès aux salons qui suivent.

Syntaxes des commandes :

Pour démontrer la manière dont sont articulées les commandes, nous utiliserons, pour les exemples de syntaxes, les informations suivantes :

- l'adresse email « **xxx@xxx.com** »
 - la durée en minutes « **60** »
 - le nom de salon « **Comptabilite** »
-

Kill :

→ Syntaxe : /kill

Exemple :

```
[ADMIN] Entrez une commande : /kill  
  
Arrêt du serveur en cours...
```

Bannissement :

→ Syntaxe : /ban xxx@xxx.com

Exemple :

```
[ADMIN] Entrez une commande : /ban xxx@xxx.com  
  
xxx@xxx.com a été banni.
```

```
MySQL localhost:3306 ssl sae_302 SQL > SELECT * FROM sanctions;  
+-----+-----+-----+-----+-----+-----+-----+  
| id_sanction | type_sanction | duree_sanction | motif_sanction | ip_client | email_client | horodatage_sanction |  
+-----+-----+-----+-----+-----+-----+-----+  
| 1 | ban | NULL | Ban administratif. | 192.168.1.2 | xxx@xxx.com | 2024-01-01 02:15:25 |  
+-----+-----+-----+-----+-----+-----+-----+
```

Révocation du bannissement :

→ Syntaxe : `/unban xxx@xxx.com`

Exemple :

```
[ADMIN] Entrez une commande : /unban xxx@xxx.com  
xxx@xxx.com a été débanni.
```

```
MySQL localhost:3306 ssl sae_302 SQL > SELECT * FROM sanctions;  
Empty set (0.0004 sec)
```

Kick :

→ Syntaxe : `/kick xxx@xxx.com 60`

Exemple :

```
[ADMIN] Entrez une commande : /kick xxx@xxx.com 60  
xxx@xxx.com a été exclu temporairement pour 60 minutes.
```

```
MySQL localhost:3306 ssl sae_302 SQL > SELECT * FROM sanctions;  
+-----+-----+-----+-----+-----+-----+-----+  
| id_sanction | type_sanction | duree_sanction | motif_sanction | ip_client | email_client | horodatage_sanction |  
+-----+-----+-----+-----+-----+-----+-----+  
| 2 | kick | 60 | Kick administratif. | 192.168.1.2 | xxx@xxx.com | 2024-01-01 02:19:03 |  
+-----+-----+-----+-----+-----+-----+-----+
```

Révocation du kick :

→ Syntaxe : `/unkick xxx@xxx.com`

Exemple :

```
[ADMIN] Entrez une commande : /unkick xxx@xxx.com  
Le kick sur xxx@xxx.com a été révoqué.
```

```
MySQL localhost:3306 ssl sae_302 SQL > SELECT * FROM sanctions;  
Empty set (0.0003 sec)
```

Grant :

→ Syntaxe : `/grant Comptabilite xxx@xxx.com`

Exemple :

```
[ADMIN] Entrez une commande : /grant Comptabilite xxx@xxx.com

xxx@xxx.com a été ajouté au salon Comptabilite.
```

```
MySQL localhost:3306 ssl sae_392 SQL> SELECT *
-> FROM membres_salons_publics
-> JOIN clients ON clients.id_client = membres_salons_publics.id_client
-> JOIN salons_publics ON salons_publics.id_salon_public = membres_salons_publics.id_salon_public;
```

id_membre	id_client	id_salon_public	id_client	nom	prenom	email	mot_de_passe	permission	id_salon_public	nom_salon	description
1	2	1	2	test	test	test@test.com	test	utilisateur	1	General	Salon par défaut.
2	3	1	3	xxx	xxx	xxx@xxx.com	xxx	utilisateur	1	General	Salon par défaut.
3	3	3	3	xxx	xxx	xxx@xxx.com	xxx	utilisateur	3	Comptabilite	Accès sur traitement de la demande.

Révocation du grant :

→ Syntaxe : `/revoke Comptabilite xxx@xxx.com`

Exemple :

```
[ADMIN] Entrez une commande : /revoke Comptabilite xxx@xxx.com

xxx@xxx.com a été retiré du salon Comptabilite.
```

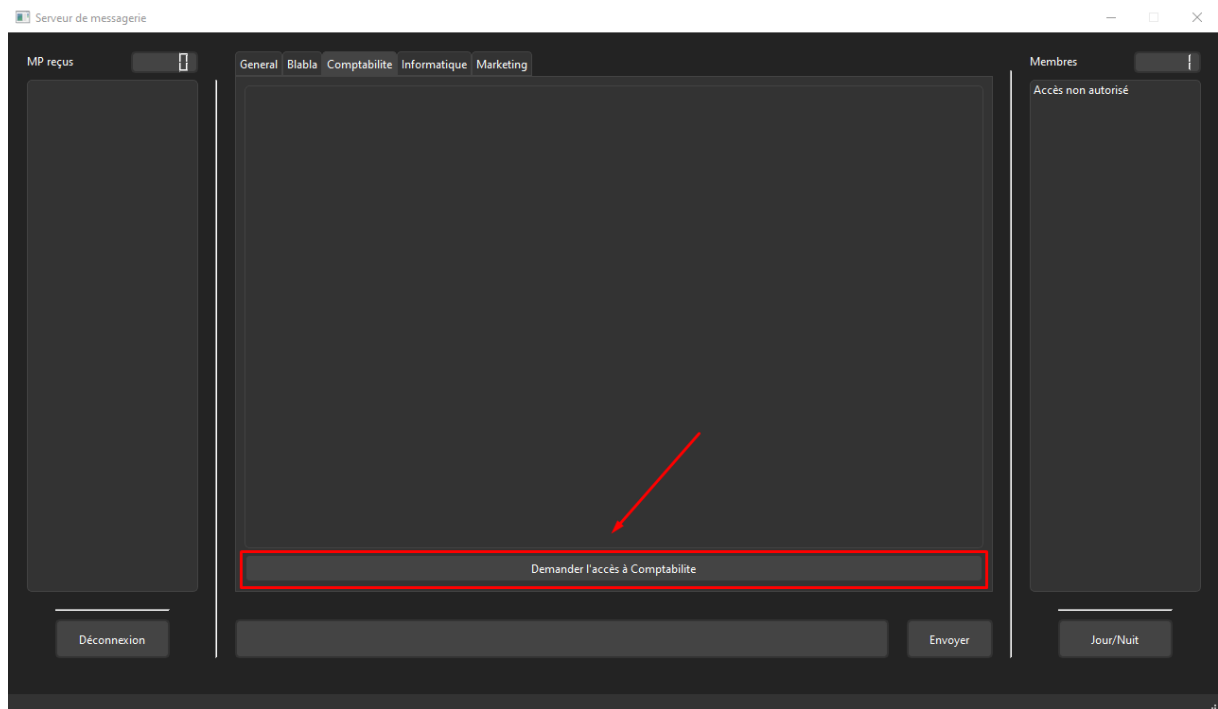
```
MySQL localhost:3306 ssl sae_392 SQL> SELECT * FROM membres_salons_publics JOIN clients ON clients.id_client = membres_salons_publics.id_client JOIN salons_publics ON salons_publics.id_salon_public = membres_salons_publics.id_salon_public;
```

id_membre	id_client	id_salon_public	id_client	nom	prenom	email	mot_de_passe	permission	id_salon_public	nom_salon	description
1	2	1	2	test	test	test@test.com	test	utilisateur	1	General	Salon par défaut.
2	3	1	3	xxx	xxx	xxx@xxx.com	xxx	utilisateur	1	General	Salon par défaut.

III - Cas spécifique :

Il existe un cas très particulier concernant les demandes d'accès au salon. En effet, les clients peuvent, via l'interface principale de messagerie, cliquer sur un bouton afin de demander l'accès à un salon.

Cela se présente de cette manière :



Lorsque le client clique sur le bouton de demande d'accès, le serveur reçoit un message de protocole et attend un input permettant via « **O** » ou « **N** » d'accorder ou de refuser immédiatement l'accès au salon :

```
[ADMIN] Entrez une commande :  
Message reçu de 192.168.1.2: [PROTOCOLE]ACCES_SALON:Comptabilite  
  
Nombre de correspondances trouvées: 0  
Accorder l'accès au salon Comptabilite à 192.168.1.2, xxx@xxx.com ? [O/N] :
```

Lorsque cela se produit, cet input empiète sur celui de la boucle de commandes administrateur, nécessitant ainsi d'effectuer deux saisies : la première permettant de répondre au reliquat de la demande d'input de commandes admin, la deuxième permettant de répondre à la demande d'accès au salon, comme suit :

```

Commande non reconnue :
[ADMIN] Entrez une commande :
Message reçu de 192.168.1.2: [PROTOCOLE]ACCES_SALON:Comptabilite

Nombre de correspondances trouvées: 0
Accorder l'accès au salon Comptabilite à 192.168.1.2, xxx@xxx.com ? [O/N] : 0

Commande non reconnue : 0
0

Nombre de correspondances trouvées: 0
[ADMIN] Entrez une commande :
Message reçu de 192.168.1.2: [PROTOCOLE]REQUETE_MEMBRES_SALONS_PUBLICS:
[["Comptabilite", "xxx xxx:xxx@xxx.com"], ["General", "test test:test@test.com,xxx xxx:xxx@xxx.com"]]

```

On observe que le premier input « **O** » est traité par la boucle de commandes administrateurs, tandis que le deuxième input « **O** » répond bien à la requête de demande d'accès en accordant l'accès à « **Comptabilite** » au client « **xxx@xxx.com** »

```
mysql [localhost:3306 sql_sae_392 SQL] - SELECT * FROM membres_salons_publics JOIN clients ON clients.id_client = membres_salons_publics.id_client JOIN salons_publics ON salons_publics.id_salon_public = membres_salons_publics.id_salon_public;
```

id_membre	id_client	id_salon_public	id_client	nom	prenom	email	mot_de_passe	permission	id_salon_public	nom_salon	description
1	2	1	2	test	test	test@test.com	test	utilisateur	1	General	Salon par défaut.
2	3	1	3	xxx	xxx	xxx@xxx.com	xxx	utilisateur	1	General	Salon par défaut.
4	3	3	3	xxx	xxx	xxx@xxx.com	xxx	utilisateur	3	Comptabilite	Accès sur traitement de la demande.

Il faut donc garder en tête que, bien que la gestion des accès concurrents sur les inputs soit à revoir, cela reste « fonctionnel ».

IV – Configuration réseau et connexion MySQL:

Par souci de simplicité et de sécurité, j'ai préféré ne pas proposer de sélection pour les informations de configuration basique de la connectivité serveur, qui devront être remplacées (si nécessaire) manuellement dans le code serveur, des lignes 1398 à 1406, comme suit :

```
1397 # Paramètres de configuration du serveur
1398 hote_init, port_init = '0.0.0.0', 24793
1399
1400 # Paramètres de connexion à la base de données
1401 mysql_init = {
1402     'host': 'localhost',
1403     'port': 3306,
1404     'user': 'serveur',
1405     'password': 'sae302',
1406     'db': 'sae_302',
1407 }
1408
```

Où :

hote_init = L'adresse IP de l'interface d'écoute du serveur

port_init = Le port utilisé par le serveur

host = le type d'hôte utilisé par l'utilisateur MySQL

port = le port utilisé pour la connexion MySQL

user = le nom de l'utilisateur de connexion à la BDD MySQL

password = le mot de passe de l'utilisateur de connexion à la BDD MySQL

db = le nom de la base de données MySQL cible

C'est tout pour la procédure d'utilisation du serveur !