

# Documentation du Protocole du Serveur de Chat

## Table des matières :

1. Introduction
  2. Commandes disponibles
    - 2.1 LOGIN
    - 2.2 MSG
    - 2.3 NEW\_MSG
    - 2.4 LOG
    - 2.5 MODIF\_MSG
    - 2.6 REMOVE\_MSG
    - 2.7 BLOCK\_CLIENT
    - 2.8 UNBLOCK\_CLIENT
    - 2.9 BAN\_CLIENT
    - 2.10 UNBAN\_CLIENT
    - 2.11 NAME
    - 2.12 LIST\_TOKENS
    - 2.13 LOGOUT
    - 2.14 EXIT
    - 2.15 TOKEN
  3. Gestion des Tokens
  4. Règles de communication entre utilisateurs
  5. Base de données
  6. Logs et journalisation
  7. Configuration du serveur
  8. Limitation des requêtes
  9. Option
-

# 1. Introduction

Ce document décrit le protocole utilisé par le serveur de chat implémenté dans le fichier serveur.c. Le serveur gère les connexions des clients, authentifie les utilisateurs via une clé API, et permet l'envoi et la réception de messages entre utilisateurs. Le serveur utilise une base de données PostgreSQL pour stocker les informations des utilisateurs, les messages, et les tokens de session.

pour l'installation et le lancement voir **README.md**

---

## 2. Commandes disponibles

### 2.1 LOGIN

- Format : LOGIN:<api\_key>
- Description : Authentifie l'utilisateur avec une clé API. Si la clé est valide, un token de session est généré et renvoyé au client.
- Réponse :
  - Succès : Connecté en tant que : <type\_utilisateur>\nVotre TOKEN:<token>
  - Échec : LOGIN FAILED

### 2.2 MSG

- Format : MSG
- Description : Récupère les messages non lus destinés à l'utilisateur connecté. Les messages sont marqués comme "reçus" après avoir été lus.
- Réponse :
  - Succès : Liste des messages avec leurs ID, émetteur, contenu, et date d'envoi.
  - Échec : Aucun message trouvé.

## 2.3 NEW\_MSG

- Format : NEW\_MSG:<id\_destinataire>,<message>
- Description : Envoie un nouveau message à un destinataire spécifié par son ID.
- Réponse :
  - Succès : Message envoyé à <destinataire> : <message>
  - Échec : Erreur : destinataire introuvable. ou Erreur : le message est trop long.

## 2.4 LOG

- Format : LOG:<id\_reference>
- Description : Récupère les messages envoyés ou reçus par l'utilisateur, avec un ID de référence optionnel pour paginer les résultats.
- Réponse :
  - Succès : Liste des messages avec leurs ID, émetteur, contenu, et date d'envoi.
  - Échec : Aucun message trouvé.

## 2.5 MODIF\_MSG

- Format : MODIF\_MSG:<id\_message>,<nouveau\_contenu>
- Description : Modifie le contenu d'un message envoyé par l'utilisateur, à condition que le message n'ait pas déjà été reçu par le destinataire.
- Réponse :
  - Succès : Message ID <id\_message> a été modifié avec succès.
  - Échec : Erreur : Vous n'êtes pas l'émetteur de ce message. Erreur : Ce message a déjà été supprimé. ou Erreur : Le message a déjà été reçu.

## 2.6 REMOVE\_MSG

- Format : REMOVE\_MSG:<id\_message>
- Description : Supprime un message envoyé par l'utilisateur, à condition que le message n'ait pas déjà été reçu par le destinataire.
- Réponse :
  - Succès : Message ID <id\_message> a été supprimé avec succès.
  - Échec : Erreur : Vous n'êtes pas l'émetteur de ce message. ou Erreur : Ce message a déjà été supprimé.

## 2.7 BLOCK\_CLIENT

- Format : BLOCK\_CLIENT:<client\_id>,<pro\_id> (Admin) ou BLOCK\_CLIENT:<client\_id> (Pro)
- Description : Bloque un client pour qu'il ne puisse pas envoyer de messages à un professionnel spécifique (Admin) ou à l'utilisateur actuel (Pro).
- Réponse :
  - Succès : Le client ID <client\_id> a été bloqué pour communiquer avec le professionnel ID <pro\_id> pendant 24 heures.
  - Échec : Erreur : Le client ID <client\_id> n'existe pas.

## 2.8 UNBLOCK\_CLIENT

- Format : UNBLOCK\_CLIENT:<client\_id>,<pro\_id> (Admin) ou UNBLOCK\_CLIENT:<client\_id> (Pro)
- Description : Débloquent un client précédemment bloqué.
- Réponse :
  - Succès : Le client ID <client\_id> a été débloquent.
  - Échec : Erreur : Le client ID <client\_id> n'est pas bloqué.

## 2.9 BAN\_CLIENT

- Format : BAN\_CLIENT:<client\_id>
- Description : Bannit un client, l'empêchant de se connecter au serveur.
- Réponse :
  - Succès : Le client ID <client\_id> a été banni avec succès.
  - Échec : Erreur : Le client ID <client\_id> est déjà banni.

## 2.10 UNBAN\_CLIENT

- Format : UNBAN\_CLIENT:<client\_id>
- Description : Débannit un client précédemment banni.
- Réponse :
  - Succès : Le client ID <client\_id> a été débanni avec succès.
  - Échec : Erreur : Le client ID <client\_id> n'est pas banni.

## 2.11 NAME

- Format : NAME
- Description : Renvoie le nom et le type de l'utilisateur actuellement connecté.
- Réponse :
  - Succès : <type\_utilisateur> connecté : <nom\_utilisateur>
  - Échec : Non connecté

## 2.12 LIST\_TOKENS

- Format : LIST\_TOKENS
- Description : Liste tous les tokens actifs (réservé aux administrateurs).
- Réponse :
  - Succès : Liste des tokens avec leurs ID, ID Compte, Token, et date d'expiration.
  - Échec : Erreur : seul un administrateur peut accéder à la liste de tokens.

## 2.13 LOGOUT

- Format : LOGOUT
- Description : Déconnecte l'utilisateur actuel.
- Réponse :
  - Succès : Déconnexion réussie.
  - Échec : Aucun compte connecté.

## 2.14 EXIT

- Format : EXIT
- Description : Ferme la connexion avec le client.
- Réponse : Aucune.

## 2.15 TOKEN

- Format : TOKEN
  - Description : Vérifie la validité du token de session actuel et renvoie le temps restant avant expiration.
  - Réponse :
    - Succès : Token valide pour l'utilisateur :  
<nom\_utilisateur>\nToken actuel : <token>\nTemps restant :  
<secondes> secondes
    - Échec : Token invalide ou expiré.
- 

# 3. Gestion des Tokens

- Génération de token : Un token est généré aléatoirement lors de la connexion d'un utilisateur. Il est valide pendant une durée définie par TOKEN\_EXPIRATION\_TIME.
  - Validation de token : Le serveur vérifie la validité du token à chaque requête nécessitant une authentification.
  - Expiration : Les tokens expirés sont automatiquement supprimés de la base de données.
-

## 4. Règles de communication entre utilisateurs

- Membres : Ne peuvent pas envoyer de messages à d'autres membres.
  - Professionnels : Ne peuvent pas envoyer de messages à d'autres professionnels.
  - Administrateurs : Peuvent envoyer des messages à tous les utilisateurs.
- 

## 5. Base de données

Le serveur utilise une base de données PostgreSQL pour stocker les informations suivantes :

- Comptes utilisateurs : `_compte`
  - Messages : `_chat_message`
  - Tokens de session : `_chat_tokensession`
  - Blocages : `_chat_bloque`
  - Bannissements : `_chat_ban`
- 

## 6. Logs et journalisation

Le serveur enregistre les événements dans un fichier log (`LOG_FILE_PATH`). Les logs incluent les connexions, les déconnexions, les erreurs, et les commandes exécutées.

---

## 7. Configuration du serveur

Le serveur est configuré via des constantes définies dans config.env, incluant :

- PORT : Port d'écoute du serveur.
- BAN\_DURATION : Durée du bannissement en secondes.
- MAX\_MESSAGES : Nombre maximum de messages à récupérer.
- MAX\_MESSAGE\_SIZE : Taille maximale d'un message.
- TOKEN\_EXPIRATION\_TIME : Durée de validité d'un token en secondes.
- ECT...

## 8. Limitation des requêtes (Rate Limiting)

Le serveur implémente un mécanisme de rate limiting pour empêcher les abus et les attaques par déni de service (DoS). Ce mécanisme limite le nombre de requêtes qu'un client peut effectuer dans un laps de temps donné.

## 9. Option

Le serveur est configuré pour recevoir 2 option :

- --help: affiche les options possibles et coupe le serveur.
- --verbose: affiche les logs dans le serveur en plus que dans le fichier.