

Request for comments

Le protocole ChatHack

Résumé :

Le protocole ChatHack est un protocole utilisé pour la communication entre clients en TCP.

Depuis un serveur, un client peut envoyer un message à tous les autres clients connectés sur ce même serveur ou établir une connexion privée avec un (ou plusieurs) client(s) connecté(s).

En connexion privée, tous les messages privés et les fichiers sont envoyés par une connexion directe entre les deux clients. Le serveur permet juste dans ce cas d'échanger les adresses des clients.

Ce document décrit le protocole et ses types de trames. Ce document explique également les raisons de certaines décisions de conception.

I. Format des données

Il y a trois formats de données différents qui constituent nos différents trames :

- Les entiers (INT) : codés sur 4 octets signés en Big Endian
- Les longs (LONG) : codés sur 4 octets signés en Big Endian
- Les chaînes de caractères (STRING) : encodées par un INT (pour la taille en nombre d'octets de la chaîne encodée en UTF-8) suivi des octets représentant la chaîne en UTF-8

Format d'un trame **STRING**

int	string	
+-----+	+-----+	
taille	Chaîne	= STRING
+-----+	+-----+	

II. Communication entre Client et Serveur :

1. Authentification

a. Côté Client

L'Authentification est la première étape du protocole de communication ChatHack entre le client et le serveur.

Le client peut s'authentifier de deux manières différentes auprès du serveur avec :

- un pseudonyme et un mot de passe
- un pseudonyme uniquement

La demande d'authentification du client se fait directement avec un code opérande **10** ou **12**.

Tant que le serveur n'a pas authentifié le client, le client ne peut envoyer que la trame d'authentification.

Il est impératif que le client soit authentifié pour qu'il puisse accéder aux différentes fonctionnalités du chat.

L'identifiant et le mot de passe ne doivent pas dépasser : **1024 octets**

la trame d'authentification est au format suivant :

- Cas n°1 : Authentification avec un pseudonyme et un mot de passe

OP	int	string	int	string
+-----+	+-----+	+-----+	+-----+	+-----+
10	Taille pseudo	Pseudonyme	Taille mot de passe	Mot de passe
+-----+	+-----+	+-----+	+-----+	+-----+

Comme dit précédemment dans la section “*Format des données*”, on formatera la trame de la manière suivante pour faciliter la lecture des différents trames :

OP	STRING	STRING
+-----+	+-----+	+-----+
10	Pseudonyme	Mot de passe
+-----+	+-----+	+-----+

- Cas n°2 : Authentification avec un pseudonyme uniquement

OP	STRING
+-----+	+-----+
12	Pseudonyme
+-----+	+-----+

b. Côté Serveur

Problèmes de connexion :

Le serveur réalise plusieurs vérifications :

- le pseudonyme n'est pas déjà pris par un autre client.
- le pseudonyme et le mot de passe concordent
- le client authentifié sans mot de passe n'utilise pas le pseudonyme d'un utilisateur de la base de données

Si le serveur accepte l'authentification du client, il pourra communiquer avec les autres clients connectés sinon le serveur refuse l'authentification et demande alors au client de se réidentifier avec des identifiants valides.

En cas d'erreur de l'authentification, le serveur renverra un frame contenant avec le code opérande **90** représentant une erreur de connexion :

OP	STRING
+-----+	+-----+
11 ou 13	Message
+-----+	+-----+

L'opérande 11 est la réponse à la demande de connexion avec mot de passe.

L'opérande 13 est la réponse à la demande de connexion anonyme.

Le message indique, s'il y a eu une erreur, ou si le client a bien été connecté.

2. Connaître les clients connectés

[Cette fonctionnalité est facultative]

a. Côté Client

Un client peut demander au serveur de lui envoyer une liste des noms des clients connectés. Cela permettra notamment au client de connaître les clients connectés et de réaliser plus tard s'il le souhaite des connexions privées avec l'un d'eux.

Un client demande la liste des noms des clients connectés avec la commande **CLIENTS** ayant pour code opérande **30** :

CLIENTS

OP	
+-----+	
	30
+-----+	

b. Côté Serveur

Le serveur reçoit la trame **CLIENTS** et renvoie la trame contenant la liste des noms des clients connectés. Ce trame a pour code opérande **30** :

OP	STRING
+-----+	
	30
	Noms des clients connectés
+-----+	

4. Discussion entre tous les clients connectés

a. Côté Client

Envoi d'un message public:

Un client peut envoyer un message à tous les autres clients connectés grâce à la commande “/” ou “@” le reste de la commande est interprété comme un message public avec un code opérande **14**.

Cette commande contient le pseudonyme du client à l'origine du message ainsi que le message que souhaite envoyer le client.

Attention le message ne doit pas excéder **1024 octets** : Si la taille est dépassée on ferme la connexion.

Le pseudonyme doit être le pseudonyme utilisé lors de l'authentification.

MSG

OP	STRING	STRING
+-----+	+-----+	+-----+
14	Pseudonyme	Message
+-----+	+-----+	+-----+

Pseudonyme : correspond à l'expéditeur du message.

Message : correspond au message à transmettre

Réception d'un message public:

Le serveur transmet le message public à tous les clients connectés. Chaque client reçoit ce message public.

Le format de la trame est donc le suivant :

OP	STRING	STRING
+-----+	+-----+	+-----+
14	Pseudonyme	Message
+-----+	+-----+	+-----+

b. Côté Serveur

Le serveur après avoir reçu la trame contenant le message, doit retransmettre ce message à tous les autres clients connectés.

la trame envoyé est de la même structure que la trame reçu. Il le retransmet à tous les clients connectés :

OP	STRING	STRING
+-----+	+-----+	+-----+
14	Pseudonyme	Message
+-----+	+-----+	+-----+

III. Communication entre Client et Client:

1. Négociation d'une connexion privée

Dans cette section, on désignera le client A voulant négocier une connexion privée avec le client B. Pour négocier cette connexion, les trames sont réceptionnées par le serveur.

Client A (initie la connexion) <--> Serveur <--> Client B

a. Côté Client

Lorsqu'un client A veut discuter en privé avec un client B, celui-ci va d'abord négocier une connexion privée en envoyant une trame qui va être réceptionnée par le serveur. Le client A va envoyer une demande de connexion avec la commande **@login + message** si, la connexion n'est n'a pas été établie (avec le code opérande **15**).

OP	STRING	STRING
15	Pseudonyme client A	Pseudonyme client B

Pseudonyme client A: expéditeur

Pseudonyme client B: destinataire

b. Côté Server

Le serveur transmet la demande du client A vers le client B avec la trame suivant ayant pour opérande le numéro 16

OP	STRING
16	Pseudonyme client B

Pseudonyme client A: expéditeur

Pseudonyme client B: destinataire

Destinataire incorrect :

Si le pseudonyme du client est incorrect, ou si le client auprès duquel on veut négocier une connexion privée n'est pas connecté alors le serveur enverra la trame d'erreur suivant :

OP	STRING
+-----+	+-----+
17	Message
+-----+	+-----+

Par exemple, le message d'erreur est : "*Bob is not connected*"

Le client reçoit la demande et lit la demande avec la trame dont l'opérande est la numéro 16.

Dès que le client B reçoit la trame par le biais du serveur, ce dernier peut soit :

- **Accepter la demande de connexion** et dans ce cas, renvoyer son accord avec la commande **/accept** avec le code opérande **18**.

OP	STRING	SOCKETADDRESS	LONG
+-----+	+-----+	+-----+	+-----+
18	Pseudonyme client A	socket adresse du client B	connectID
+-----+	+-----+	+-----+	+-----+

- **Refuser la demande de connexion**, et dans cet autre cas, exprimer son désaccord avec la commande **/refuse** et un code opérande **19**.

OP	STRING
+-----+	+-----+
19	Pseudonyme client A
+-----+	+-----+

la trame est alors renvoyé au client A et contient un message précisant la raison de ce refus de connexion.

- **Aucune réponse du client "Facultatif"**: la demande de connexion expire dans le cas ou aucune réponse n'est donnée par le client. Le serveur se charge d'informer les clients concernés.

La réception de la trame a pour code opérande **25** :

OP	STRING
+-----+	+-----+
25	Session Expirée
+-----+	+-----+

Une fois la demande de connexion privée accepté, une demande de vérification est réalisé. Le client qui est à l'origine de la demande et qui reçoit le connect_id de l'expéditeur renvoi une trame de vérification au destinataire.

OP	STRING	LONG
+-----+	+-----+	+-----+
20	login	connect_id
+-----+	+-----+	+-----+

Le destinataire recoit le connect_id, verifie qu'ils sont identiques et renvoie a son tour un message de confirmation :

OP	STRING
+-----+	+-----+
21	login
+-----+	+-----+

2. Discussion entre deux clients privés

a. Envoi d'un message client-client

En connexion TCP directe, pour envoyer un message du client A vers le B (*ou l'inverse*) le client utilisera la commande **@login message** avec le code d'opérande **23**. Le client utilise le jeton unique de connexion envoyée par le serveur au moment de la négociation de connexion. L'utilisation du jeton connexion privée permet de savoir à qui le client souhaite envoyer le message. On rappelle que la taille du message ne peut excéder **1024 octets**.

OP	STRING	STRING
+-----+	+-----+	+-----+
23	login	message
+-----+	+-----+	+-----+

b. Envoi d'un fichier client-client

Si le client A souhaite envoyer un fichier au client B celui-ci utilisera la commande **/login file**, avec le code opération **22**. De même que pour l'envoi d'un message le client utilise le jeton unique de connexion privée. L'envoi du fichier est basé sur TCP.

OP	STRING	STRING	BYTEBUFFER
+-----+	+-----+	+-----+	+-----+
22	login	fichier_name	bytebuffer_file
+-----+	+-----+	+-----+	+-----+