

Documentation protocole et filtres SYR 2

Une requête entre le serveur et le client à cette forme :

```
typedef struct req {  
    int typeReq;  
    int id;  
    int tailleData;  
    char data[R_tailleMaxData];  
} requete;
```

typeReq : définit le type de requête (ex : demande de connexion, envoi fichier son ...).
Utiliser les constantes pour l'initialiser.

id : identifiant d'un client permettant au serveur de distinguer les clients. Si c'est un client sans id attribué ou le serveur qui effectue la requête, id doit être égal à R_idNull.

tailleData : indique la taille du buffer de donnée.

data : correspond aux données qui seront envoyées (ex : une partie du fichier audio envoyer par le serveur au client).

Types de requêtes et leurs rôles :

R_demandeCo	Demande au serveur si le client peut se connecter et recevoir un identifiant. Envoyée par le client id non utile data non utile Réponse attendue : R_okDemandeCo R_serverPlein
R_okDemandeCo	Requête envoyée par le serveur pour confirmer que le client peut se connecter. Envoyée par le serveur id non utile data obligatoire, contient l'id attribué au client Réponse attendue : Aucune
R_fermerCo	Le client informe le serveur qu'il ferme la connexion . Envoyée par le client id obligatoire data non utile Réponse attendue : R_okFermerCo R_idInexistant
R_okFermerCo	Le serveur prévient le client qu'il a bien reçu sa requête de fermeture de connexion. Envoyée par le serveur id non utile data non utile Réponse attendue : Aucune
R_demanderFichierAudio	Demande au serveur de commencer à lire un fichier. Envoyée par le client id obligatoire data obligatoire, contient le nom du fichier à ouvrir Réponse attendue : R_okDemanderFichierAudio R_fichierAudioNonTrouver R_idInexistant

R_okDemanderFichierAudio	<p>Réponse du serveur si le fichier demandé a été trouvé. Envoyée par le serveur</p> <p>id non utile</p> <p>data obligatoire, contient des informations sur le fichier son : fréquence, taille, mono ou stéréo</p> <p>Réponse attendue:</p> <p>Aucune</p>
R_fichierAudioNonTrouver	<p>Réponse du serveur si le fichier demandé n'a pas été trouvé ou n'est pas un fichier son. Envoyée par le serveur</p> <p>id non utile</p> <p>data non utile</p> <p>Réponse attendue :</p> <p>Aucune</p>
R_demandePartieSuivanteFichier	<p>Le client demande au serveur de lui envoyer la suite du fichier. La partie data doit contenir le numéro du bloc de fichier permettant ainsi de vérifier si il y a eu une perte de paquet lors de la transmission. Si le serveur a le même numéro que celui envoyé par le serveur c'est qu'il y a eu une perte lors de l'envoi serveur → client. Si le serveur a pour numéro, (numéro_client - 1), tout s'est bien passé et le serveur envoie la suite du fichier.</p> <p>Envoyée par le client</p> <p>id obligatoire</p> <p>data obligatoire</p> <p>Réponse attendue :</p> <p>R_okPartieSuivanteFichier</p> <p>R_finFichier</p> <p>R_fichierAudioNonTrouver</p> <p>R_idInexistant</p>
R_okPartieSuivanteFichier	<p>Le serveur confirme qu'il a bien reçu la demande de lecture du fichier. Envoyée par le serveur</p> <p>id non utile</p> <p>data obligatoire, contient une partie du fichier demandé</p> <p>Réponse attendue :</p> <p>Aucune</p>
R_finFichier	<p>Indique au client que le serveur a envoyé tout le fichier.</p> <p>Envoyée par le serveur</p> <p>id non utile</p> <p>data non utile</p> <p>Réponse attendue :</p> <p>Aucune</p>
R_serverPlein	<p>Requête qu'envoie le serveur pour prévenir le client qu'il n'y a plus de place. Envoyée par le serveur</p> <p>id non utile</p> <p>data non utile</p> <p>Réponse attendue :</p> <p>Aucune</p>
R_idInexistant	<p>Requête qu'envoie le serveur si il reçoit un id qui n'est pas défini. Envoyée par le serveur</p> <p>id non utile</p> <p>data non utile</p> <p>Réponse attendue :</p> <p>Aucune</p>

Filtres :

Filtre sur la modification du volume d'écoute

- vol suivi d'un entier modifie le volume gauche et droite
- gauche suivi d'un entier modifie le volume gauche.
- droite suivi d'un entier modifie le volume droit.

Par défaut, le volume est à 100%. Si le son est en mono, alors le volume sera la moyenne du coté gauche et droite

Modification de la vitesse de lecture

- vit suivi d'un entier, par défaut la vitesse est à 100%

Lecture en mode mono

- mono force à jouer la musique en mono

Lecture avec écho

-echo permet d'ajouter de l'écho au fichier audio avec un décalage de 0.5 seconde, et une baisse de son de 25 %

Lecture inverse/symétrique

- inv va inverser le coté gauche et droite si le son est en stéréo

Il est possible de lancé plusieurs filtre en même temps, par exemple pour mettre de l'écho et baissé le volume du coté gauche de moitié :

padsp bin/audioclient ip fichier -echo -gauche 50

L'ordre des argument n'a pas d'importance dans ce cas, en revanche dans les cas ou certain paramètre rentre en conflit, c'est le dernier argument qui est prioritaire. Exemple :

padsp bin/audioclient ip fichier -vol 150 -gauche 25 -gauche 10

cela aura pour effet de mettre le volume de droite à 150 % et le volume gauche à 10 %