

Td 1 - Node.js v0.10.33 - Irc like App

16 octobre 2017

1

Stream et Socket TCP

L'objectif de ce td est de concevoir une application similaire à IRC, application permettant l'échange de messages entre utilisateurs. Plus précisément, vous devez créer un client de messagerie et son serveur associé *via* l'utilisation des primitives de node.js pour manipuler les **Stream**, les documents **Json** et les **sockets** TCP.

1. Le client et le serveur sont développés en node.js. Les seuls packages autorisés sont : **net**, et **minimist**.
2. Implémenter les fonctions suivantes :
 - **send** envoie un message à une personne particulière qui est connectée ou à un groupe auquel on appartient.
 - **broadcast** envoie un message à toutes les personnes connectées.
 - **list** demande au serveur de lister le nom de toutes les personnes connectées, ou uniquement celle d'un groupe.
 - **join** permet de se joindre à un groupe
 - **part** permet de quitter un groupe
 - **quit** permet de quitter proprement l'utilisateur

Chacune des commandes peut avoir des **alias** au niveau du client. Par exemple :

- **b;salut** pour broadcaster le message à toute personne connectée.

Base revision , , .

- ▣ `s;xavier;salut` envoie le message privé `salut` à l'utilisateur `xavier`.
 - ▣ `s;#node;salut` envoie un message à tous les utilisateurs du groupe `node`.
3. L'échange de messages se fait par l'envoi d'objets javascript natifs (c-a-d des documents JSON). Par exemple, la commande `b;hello` génère le message suivant reçu par le serveur : `{"name":"david","msg":"hello","action":"broadcast"}`
 4. L'utilisateur est automatiquement averti quand une personne : se connecte, se déconnecte, joint ou quitte un groupe.
 5. Voici plusieurs captures d'écrans décrivant le fonctionnement du programme.

```
app1$ node client2 --name david
local> connected
Hello david
local>b;hello
local>
```

FIGURE 1 – Broadcast d'un message

```
app1$ node client2 --name david
local> connected
Hello david
local>server>floreale is connected
local>server>xavier is connected
local>list
local>server>floreale
local>server>xavier
local>server>xavier left the chat.
```

FIGURE 2 – Notifications

```
app1$ node client2 --name david
local> connected
Hello david
local>list
local>server>floreale
local>server>xavier
local>s;floreale;salut
```

FIGURE 3 – Message privé

```
app1$ node client2 --name floreal  
local> connected  
Hello floreal  
local>david>salut  
local>
```

FIGURE 4 – Reception Message privé