Fiche technique

Description de Whatsup

**Description**

[…]

**Table des matières**

[1. Architecture 4](#_Toc41957499)

[2. SERVEUR 4](#_Toc41957500)

[2.1. Lancer le GUI serveur 5](#_Toc41957501)

[2.2. Configurer une base de donnée 5](#_Toc41957502)

[2.3. Configurer le port d’écoute et lancer le serveur 5](#_Toc41957503)

[2.4. Le nombre de clients connectés 6](#_Toc41957504)

[2.5. Les logs 6](#_Toc41957505)

[3. CLient 7](#_Toc41957506)

[3.1. Lancer le GUI client 7](#_Toc41957507)

[3.2. Joindre le serveur 8](#_Toc41957508)

[3.3. S’authentifier 8](#_Toc41957509)

[3.3.1 S’inscrire 8](#_Toc41957510)

[3.3.2 Se connecter 8](#_Toc41957511)

[3.4. Discussion 8](#_Toc41957512)

[3.4.1 Envoyer un message à un utilisateur connecté 9](#_Toc41957513)

[3.4.2 Envoyer un message à tous les utilisateurs connectés 9](#_Toc41957514)

[3.4.3 Envoyer un message à un groupe 9](#_Toc41957515)

[3.5. Partage de fichier 10](#_Toc41957516)

[3.5.1 Envoyer un fichier 10](#_Toc41957517)

[3.5.2 Telecharger un fichier 10](#_Toc41957518)

[3.6. Groupe 10](#_Toc41957519)

[3.6.1 Créer groupe 10](#_Toc41957520)

[4. BASE DE DONNEE 11](#_Toc41957521)

[5. design pattern 11](#_Toc41957522)

[5.1. Singleton 11](#_Toc41957523)

[6. Les principes SOLID 12](#_Toc41957524)

[7. Tests 12](#_Toc41957525)

[Utilisez les styles pour définir vos titres / Mettez à jour la table des matières en vous plaçant dans la table et pressez F9 ]

Architecture

L’architecture de whatsup a été un choix décisif dans la conception de celui-ci, au vu de problématique et l’objectif de l’application, nous avons décidé de nous tourner vers une architecture en deux grandes couches

* Une Couche CLIENT/SERVEUR
* Une Pseudo-MVC

Ce choix nous a permit de séparer l’application en deux projets indépendants, Un projet CLIENT et un projet Serveur, et chaqu’un de ses projets est basé sur une architecture Pseudo-MVC.

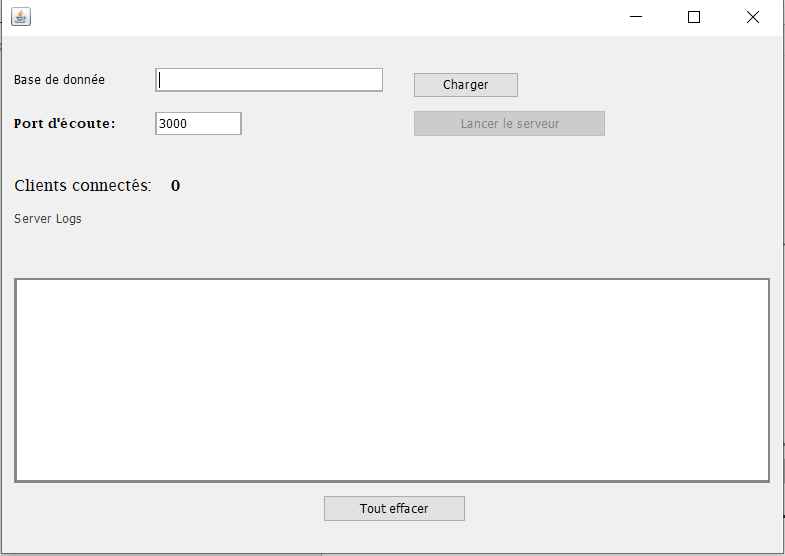
L’interaction entre ces deux parties ce fait via des appels réseaux.

Ce choix d’architecture était selon nous l’une des meilleurs facon d’implémenter whatsup, on pouvait facilement se répartir les taches, la maintenance applicative était beaucoup plus simples.

L’un des points fort de ce choix est le fait que l’on pouvait partager seulement la partie CLIENT de notre application aux utilisateurs.

SERVEUR

Le serveur est le coeur de notre application, c’est celui-ci qui va gérer la logique des échanges entre différents utilisateurs et groupes.



Lancer le GUI serveur

Exécuter L’application “Server.jar” Contenu dans le dossier JAR FILE que vous avez préalablement télécharger.

Une fois Executé L’interface d’administration du serveur s’ouvrira.

Configurer une base de donnée

Lorsque vous êtes sur le GUI SERVEUR, la prèmiere étape constiste à configurer une Base de donnée de type XML.

* Cliquer sur le Bouton charger,
* Une fois le FileManager apparu, rendez vous sur le dossier ‘JAR FILE”, puis choissisez le fichier Database.xml



La base sera automatiquement prise en compte, vous pourrez passer à l’étape suivante

Configurer le port d’écoute et lancer le serveur

Le port d’écoute du serveur, par défaut est “3000”, La modification de ce dernier est possible au niveau du champs “Port d’écoute”.



Une fois le port configuré vous pouvez cliquer sur le bouton “Lancer le serveur”



Dès lors que le serveur est lancé, une log apparaitra dans la partie “Server logs” en bas de la page

Vous indiquant que le serveur est lancé et prêt à accueillir des clients

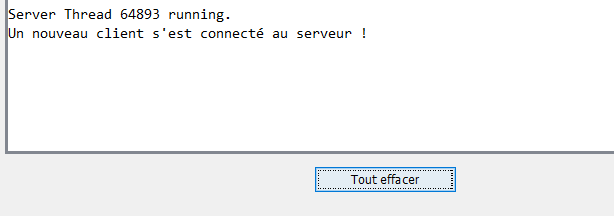


Le nombre de clients connectés

Vous avez la possibilité de voir le nombre de clients connectés au niveau de “Clients connectés” au milieu droit de la page.



Lorsqu’un nouveau client se connecte une log apparait dans “Server logs”



Les logs

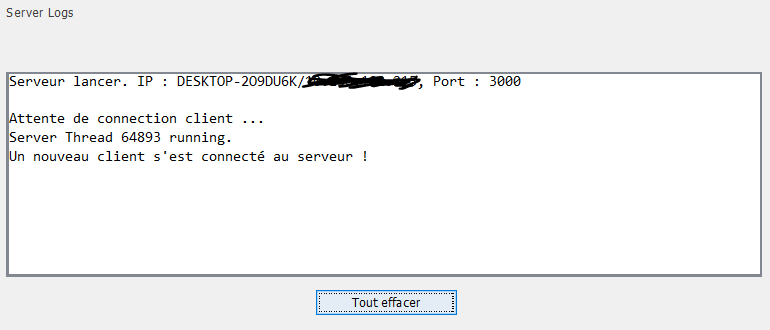
Dans la partie bas de notre application se trouve le logger, dans lequel vous pourrez être notifier des différents évenements qui se déroulent au sein du serveur :

Lancement du serveur

Lancement d’un Thread

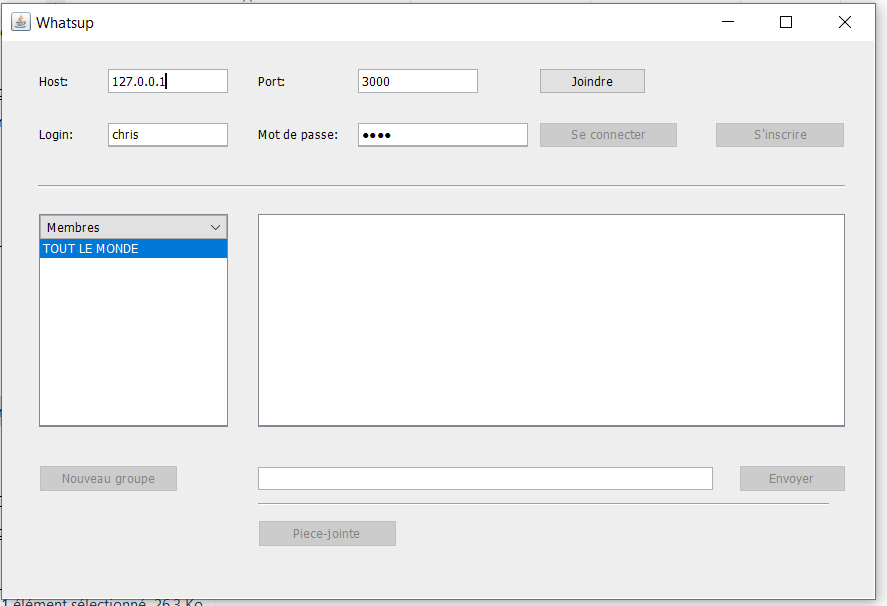
Connexion d’un client

…



CLient

La partie Client, est une l’application indépendante dédiée aux utilisateurs finaux, c’est elle qui permettra à un utilisateur d’échanger avec d’autres, en communiquant avec le serveur via des appels réseaux.



Lancer le GUI client

Une fois que le serveur Lancer;

Exécuter L’application “Client.jar” Contenu dans le dossier JAR FILE que vous avez préalablement télécharger.

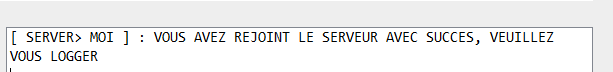
Une fois Executé L’interface Client s’ouvrira.

Joindre le serveur

Pour se connecter au serveur, il faut entrer les Host du serveur ainsi que le port, au niveau des champs “Host” et “Port” situés en haut de la fenêtre, puis cliquer sur joindre.



Une log apparaitra si le serveur a été attaint avec succès, dans le logger en Bas à gauche



S’authentifier

Une fois que le serveur rejoint vous pouvez vous authentifier, en remplissant “Login” et “Mot de passe”, en haut de la fenêtre.



Vous aurez alors deux actions disponibles “Se connecter” et “S’inscrire”

S’inscrire

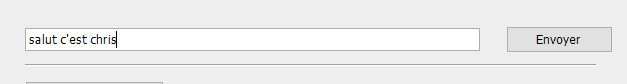
Lorsque vous cliquer sur le bouton “s’inscrire”, si le login est disponible, un nouveau compte sera créé, vous pourrez alors commencer à discuter.

Se connecter

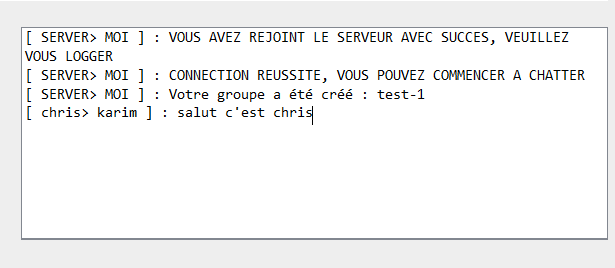
Si vous avez déjà un compte cliquer sur “Se connecter”, si les identifiants sont correctes vous pourrez alors commencer à discuter

Discussion

Vous avez la possibilité de chatter en entrant votre message dans le champs situé en bas de la fenêtre et l’envoyer en un destinataire selectionné en cliquant sur le bouton “Envoyer”

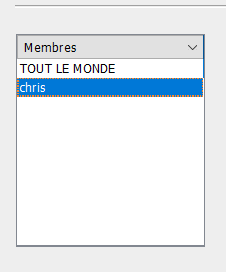


Une fois le message envoyé vous verrez la trace sur la console de log



Envoyer un message à un utilisateur connecté

Pour envoyer un message à une autre utilisateur, selectionner “Membres” au niveau de la liste déroulante à gauche de la fenêtr,e puis cliquer sur le membres désiré et envoyer votre message.

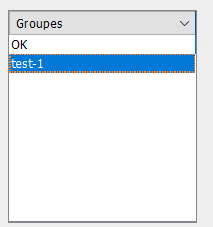


Envoyer un message à tous les utilisateurs connectés

Selectionner “TOUT LE MONDE” dans la liste des membres puis envoyer votre message

Envoyer un message à un groupe

Pour envoyer un message à Goupe, selectionner “Groupes” au niveau de la liste déroulante à gauche de la fenêtre, puis cliquer sur le groupe désiré et envoyer votre message.



Partage de fichier

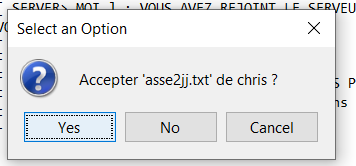
Envoyer un fichier

Vous avez la possibilité de partager des fichiers seulement entre membres, cliquer sur le membre destinataire, puis sur le bouton “pièce jointe” en bas de la fenêtre, selectionner un fichier, un bouton “envoie PJ” apparaitra, cliquer dessus pour transferer le fichier.



Telecharger un fichier

Lorsque vous recevez un fichier d’un autre membre, une pop-up apparait sur votre fenêtre, qui vous demande d’accepte ou non le Fichier



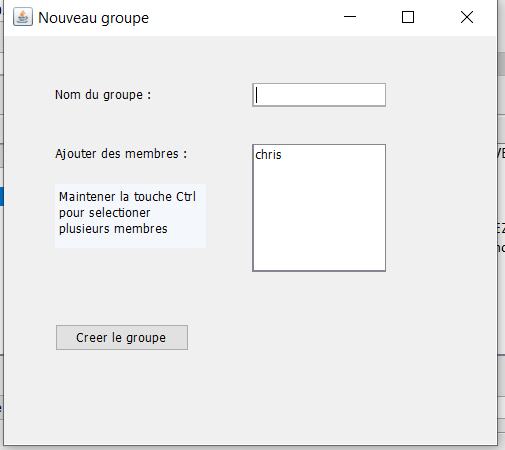
Si vous cliquer sur “Yes”, l’explorateur de fichier s’ouvrira, puis selectionner le dossier de destination et cliquer sur “SAVE”, le fichier est alors téléchargé

Groupe

un groupes est un ensemble d’utilisateurs, qui peuvent communiquer seulement entre eux

Créer groupe

* + Cliquer sur le bouton “Nouveau groupe” en bas à gauche de la fenêtre
  + Une pop-up s’ouvrira



* + Entrer le nom du groupe puis selectionner les membres
  + Cliquer “Créer le groupe”

Votre groupe et créé vous pouvez commencer à échanger avec ce groupe

BASE DE DONNEE

La Base de donnée de notre application est un fichier XML, contenant les identifiants des utilisateur et les groupes créés.

Elle est necessaire au bon fonctionnement de notre app

design pattern

Whatsupp bénéficie de l’implémentation de Design pattern, afin de permettre de répondre à des problématiques techinques.

Singleton

Nous avons utilisé le Design Pattern SINGLETON, pour l’implémentation de classe Database.java afin de n’avoir qu’une seule instance de notre Database, celà évite des accès concurrentes au fichier Database.xml doncc minimiser les conflits pouvant être liés à cette concurence

/\*\*

\* Gestion de la base de donnée XML

\* Singleton

\* **@author** Christophe Kede

\*

\*/

**public** **class** Database {

**...**

**private** **static** Database INSTANCE = **null**;

**.. .**

**public** **static** Database getInstance(String filePath)

{

**if** (INSTANCE == **null**)

{ INSTANCE = **new** Database(filePath);

}

**return** INSTANCE;}

}

Les principes SOLID

La responsabilité unique est respectée dans notre class Logique Wserver.java, Database.java, ..

Le principe de ségregration est respectée

Tests

Nous avons introduit une partie sur l’application basée sur Junit.