Module Connexion

# Introduction

Lorsque l'utilisateur ouvre le programme Chator, la première chose qu'il voit est la fenêtre de connexion. Après avoir s'être inscrit, il peut se connecter au serveur de chat à tout moment à l'aide de ces identifiants.

Dans ce document, nous allons traiter de l'implémentation de la connexion, dans le sens d'authentification auprès du serveur de chat. Ce module est étroitement lié au module d'inscription, il est donc nécessaire de se référer à ce dernier pour comprendre pleinement le fonctionnement de ce module.

# Acteur du module

## ControllerUser

Ce contrôleur est partagé avec le module d'inscription. Il permet en plus de son rôle vis-à-vis de l'inscription, de connecter la vue avec le reste du programme. Il permet la récupération des données saisies dans l'interface de connexion et l'interaction avec le serveur.

## ViewUser

La vue utilisateur a un rôle mineur. Elle est chargée de vérifier que tous les champs sont remplis lorsque l'utilisateur veut se connecter, si c'est le cas elle envoie un signal au contrôleur.

Elle se charge également d'afficher les potentielles messages d'erreurs.

# Détails sur l'interface graphique

L'interface utilisateur est très simple, à même titre que celle pour l'inscription.

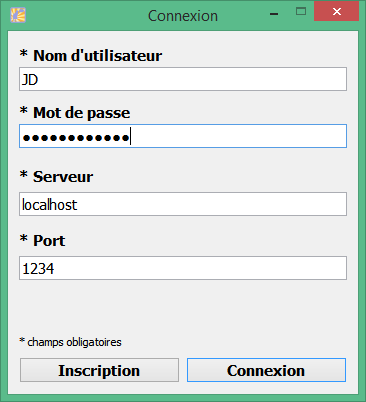


Figure Fenêtre de connexion

# Validation des champs

La seule vérification faite au niveau de la vue est la présence de tous les champs. Si l'utilisateur n'a pas mentionné un des champs, le message suivant lui sera affiché :



# Cheminement pour la connexion

## Récupération des informations depuis le formulaire

Dans un premier temps, au clic sur le bouton "Connexion", la vue va envoyer un signal au contrôleur afin de lui informer que les informations sont prêtes à être récupérées. Le contrôleur va ensuite prendre le relai.

## Ouverture d'un socket avec le serveur

Tel que décrit dans la documentation sous "Module Inscription", une connexion est tout d'abord ouverte avant l'interaction avec le serveur. Cette connexion consiste en l'ouverture d'un canal de communication avec le serveur à l'aide d'un socket. Il faut la différencier de la connexion à proprement parler au serveur de chat qui englobe le tout (vérification des identifiants, autorisation d'accès au chat).

Si l'utilisateur a déjà un socket ouvert avec le serveur, par exemple s'il avait ouvert par erreur la fenêtre d'inscription, la connexion est conservée.

## Demande du sel auprès du serveur

Une fois la connexion établie entre le serveur et le client. Une demande de sel est formulée envers le serveur. Une fois le sel reçu, l'utilisateur peut hacher son mot de passe.

## Envoi des identifiants

On peut désormais prouver son identité auprès du serveur en envoyant son nom d'utilisateur et son mot de passe haché. Le serveur va comparer le hash avec celui contenu dans sa base de données. En effet, pour des raisons de sécurité évidente, le mot de passe n'est pas stocké en clair dans la base de données.

## Récupération de la réponse du serveur

Si les informations fournies par l'utilisateur sont correctes, l'utilisateur reçoit les informations dont il a besoin : son ModelUser contenant ses détails du profil, sa paire de clé ainsi que le sel pour le chiffrement de sa clé privée.

En effet, les clés ont été générées à l'inscription par l'utilisateur puis stockée sur le serveur. Avant ce stockage, il a été nécessaire de chiffré la clé privée de l'utilisateur à l'aide du sel et de son mot de passe. Il peut désormais déchiffrer sa clé privée qui lui permettra à son tour de déchiffrer les messages qui lui sont destinés dans l'application.

Dans le cas où l'utilisateur fournit des informations erronées (nom d'utilisateur ou mot de passe), le serveur lui enverra un message d'erreur.

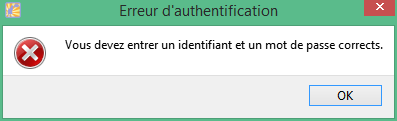


Figure Erreur lorsque l'utilsiateur fournit de mauvais identifiants.

## Ouverture du chat

Maintenant que l'utilisateur a tout en main, il peut accéder à la fenêtre de chat et utiliser l'application à sa guise.