

# DOCUMENTATION UTILISATEUR - ADINAT

*Delattre Marion - Vovc Dan-Gabriel*

## PRÉSENTATION ET OBJECTIFS

Voici notre documentation utilisateur pour le projet ADINAT (A Dog Is Not A Tchat). Ce projet repose principalement sur le développement d'une relation client-serveur de chat basé sur une RFC réalisée avant l'implémentation. Ce projet à pour objectif de respecter cette RFC en utilisant les commandes personnalisées que nous avons créées pour la communication entre le client et le serveur.

## INSTALLATION

L'installation du chat ADINAT nécessite plusieurs étapes pour s'assurer de son bon fonctionnement et puisse être correctement exécutée.

### I - Avec exécutable

#### Sous Windows

1. Prérequis
  - Python3 doit être installé sur votre système.
  - Pip doit être installé sur votre système.
2. Installation
  - Les bibliothèques pyyaml et tkinter doivent être installées sur votre système.
  - commande d'installation : **pip install pyyaml tkinter**
3. Décompresser le dossier ADINAT\_Turquoise\_Delattre\_Vovc.zip.  
Vérifier que vous avez bien les dossiers Windows, Linux, Code source, Doc.  
Accédez au dossier **Windows** et assurez vous que vous avez bien les fichiers

suivants:

Name	Date modified	Type	Size
downloads	17/04/2023 10:48	File folder	
adinat_config.yaml	17/04/2023 10:52	Yaml Source File	1 KB
client_on_Windows.exe	17/04/2023 11:51	Application	10,681 KB
server.log	17/04/2023 10:49	LOG File	0 KB
server_on_Windows.exe	17/04/2023 11:54	Application	7,742 KB

Ces fichiers sont indispensables pour la bonne exécution de l'application.

#### 4. Exécution

- **Merci de lire la partie PARAMÉTRAGE de cette documentation afin d'avoir une meilleure vision sur la configuration de l'application.**
- Désactiver Windows Defender et **lancez le serveur en premier** en double cliquant sur son exécutable (server\_on\_Windows.exe)
- Une fois le serveur en marche, double cliquez sur l'exécutable du client (client\_on\_Windows.exe)

### Sous Linux

#### 1. Prérequis

- Python3 doit être installé sur votre système.
- Pip doit être installé sur votre système.

#### 2. Installation

- Les bibliothèques pyyaml et tkinter doivent être installées sur votre système.
- Commande d'installation : **sudo pip install pyyaml tkinter**
- Alternativement il est possible d'installer tkinter avec la commande suivante : **sudo apt install python3-tk**

#### 3. Décompresser le dossier ADINAT\_Turquoise\_Delattre\_Vovc.zip.

Vérifier que vous avez bien les dossiers Windows, Linux, Code source, Doc.

Accédez au dossier **Linux** et assurez vous que vous avez bien les fichiers suivants :

Name	Date modified	Type	Size
downloads	17/04/2023 12:18	File folder	
adinat_config.yaml	17/04/2023 10:43	Yaml Source File	1 KB
client_on_Linux	17/04/2023 12:30	File	11,181 KB
server.log	17/04/2023 10:49	LOG File	0 KB
server_on_Linux	17/04/2023 12:29	File	6,567 KB

Ces fichiers sont indispensables pour la bonne exécution de l'application.

#### 4. Exécution

- **Merci de lire la partie PARAMÉTRAGE de cette documentation afin d'avoir une meilleure vision sur la configuration de l'application.**
- Double-cliquez sur l'exécutable du serveur **en premier** (server\_on\_Linux) (rien ne s'affiche et cela est normal)
- Double-cliquez sur l'exécutable du client (client\_on\_Linux)
- Lorsque vous lancez le client, une fenêtre devrait apparaître et vous pourrez ensuite interagir avec l'application.
- **Il est possible d'ouvrir un terminal et de lancer chaque exécutable individuellement en faisant la commande : `./server_on_Linux` ou `./client_on_Linux`**

## II - Sans un exécutable

### Sous Windows

#### 1. Prérequis

- Python3 doit être installé sur votre système.
- Pip doit être installé sur votre système.

#### 2. Installation

- Les bibliothèques pyyaml et tkinter doivent être installées sur votre système.
- commande d'installation : **`pip install pyyaml tkinter`**

#### 3. Décompresser le dossier ADINAT\_Turquoise\_Delattre\_Vovc.zip.

Vérifier que vous avez bien les dossiers Windows, Linux, Code source, Doc.

Accédez au dossier **Code source** et assurez vous que vous avez bien les fichiers

suivants:

Name	Date modified	Type	Size
downloads	17/04/2023 10:48	File folder	
adinat_config.yaml	17/04/2023 11:13	Yaml Sour...	1 KB
client.py	17/04/2023 12:24	Python So...	21 KB
server.py	17/04/2023 12:25	Python So...	50 KB
user.py	16/04/2023 17:02	Python So...	3 KB
server.log	17/04/2023 10:49	LOG File	0 KB

Ces fichiers sont indispensables pour la bonne exécution de l'application.

#### 4. Exécution

- **Merci de lire la partie PARAMÉTRAGE de cette documentation afin d'avoir une meilleure vision sur la configuration de l'application.**
- Dans un terminal, déplacez vous vers le dossier 'Code source' contenant les fichiers ci-dessus.
- **Dans un premier temps**, lancez la commande : `python ./server.py`
- **Dans le terminal**, vous devriez avoir le message *"The server is now running."*
- Dans un deuxième terminal, lancez la commande : `python ./client.py`
- Lorsque vous lancez le client, une fenêtre devrait apparaître et vous pourrez ensuite interagir avec l'application.

### Sous Linux

#### 1. Prérequis

- Python3 doit être installé sur votre système.
- Pip doit être installé sur votre système.

#### 2. Installation

- Les bibliothèques pyyaml et tkinter doivent être installées sur votre système.
- Commande d'installation : `sudo pip install pyyaml tkinter`
- Alternativement il est possible d'installer tkinter avec la commande suivante : `sudo apt install python3-tk`

#### 3. Décompresser le dossier ADINAT\_Turquoise\_Delattre\_Vovc.zip.

Vérifier que vous avez bien les dossiers Windows, Linux, Code source, Doc.  
Accédez au dossier **Code source** et assurez vous que vous avez bien les fichiers suivants :

Name	Date modified	Type	Size
downloads	17/04/2023 10:48	File folder	
adinat_config.yaml	17/04/2023 11:13	Yaml Sour...	1 KB
client.py	17/04/2023 12:24	Python So...	21 KB
server.py	17/04/2023 12:25	Python So...	50 KB
user.py	16/04/2023 17:02	Python So...	3 KB
server.log	17/04/2023 10:49	LOG File	0 KB

Ces fichiers sont indispensables pour la bonne exécution de l'application.

#### 4. Exécution

- **Merci de lire la partie PARAMÉTRAGE de cette documentation afin d'avoir une meilleure vision sur la configuration de l'application.**
- Dans un terminal, déplacez vous vers le dossier 'Code source' contenant les fichiers ci-dessus.
- Transformez les fichiers 'client.py' et 'server.py' en exécutable avec les commandes : `chmod +x server.py` et `chmod +x client.py`
- **Dans un premier temps**, dans un terminal, lancez le serveur avec la commande : `./server.py`
- Dans un autre terminal, lancez le client avec la commande : `./client.py`
- **Alternativement, si ce processus ne marche pas, vous pouvez lancer le serveur et le client avec les commandes : `python3 ./server.py` et `python3 ./client.py`**

## PARAMÉTRAGE

Tous les éléments de paramétrages sont présents dans le fichier `adinat_config.yaml`.

**Attention :** Les exécutable (serveur et client) lisent tous les deux dans le fichier `adinat_config.yaml`. Il faut s'assurer que ce fichier est bien à la même racine (dans le même dossier) avec les exécutables.

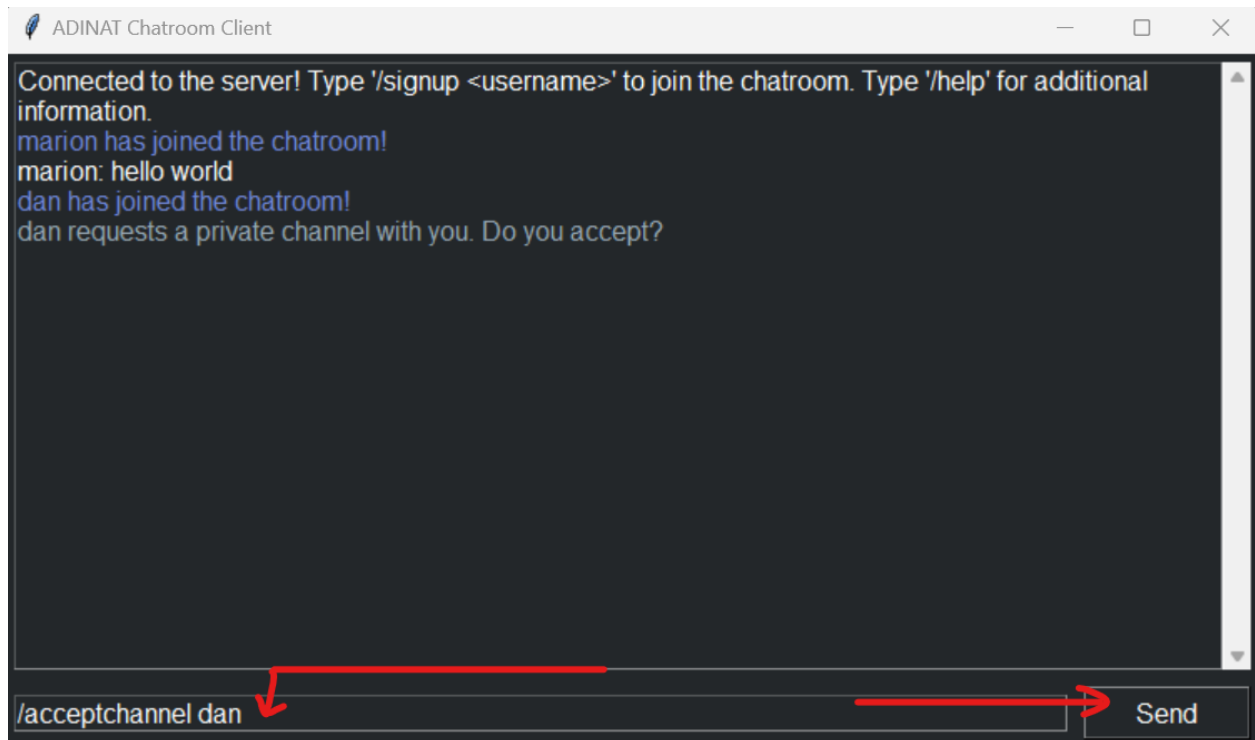
De base, le configurateur de l'application (`adinat_config.yaml`) est paramétré afin que l'application s'exécute en localhost. Cependant, vous pouvez changer les données afin qu'il s'exécute en LAN.

```
server:
  host: ""
  port: 10503
log:
  file: server.log
client:
  host: 127.0.0.1
  port: 10503
downloads: ./downloads/
```

- L'exécutable du serveur va lire les données concernant le champ '**server**' et '**log**'.
- L'exécutable du client va lire les données concernant le champ '**client**'.
- '**file**' est le nom du fichier qui contiendra les logs des commandes et réponses effectuées par le serveur. Lorsque vous changez son nom, un autre fichier va se créer s'il n'existe pas déjà.
- '**host**' est l'adresse IPv4 du serveur.
- '**port**' est le port sur lequel le serveur va écouter.
- Il s'agit bien de mettre les mêmes informations pour le '**host**' et le '**port**' pour le serveur ET le client.
- '**downloads**' est le chemin vers le dossier qui contiendra les fichiers téléchargés depuis d'autres clients. Si on souhaite changer ce chemin, il faut s'assurer que ce chemin existe.

## UTILISATION DU TCHAT

### I - Interface



L'interface est composée de 3 parties. Une fenêtre où s'affiche les messages, les erreurs, et les résultats de commandes. Juste en dessous, vous avez un champ de saisie dans lequel vous pourrez écrire vos commandes en respectant une certaine syntaxe :

- Les commandes (décrites dans la suite de la documentation) devront être précédées de l'élément “ / ”, par exemple : “/signup”.
- **Les messages envoyés à tous les utilisateurs (commande ‘/msg’) ne nécessitent pas d’être précédés d’une commande spécifique. Autrement dit, vous entrez votre message et vous envoyez tel quel.**

Puis vous avez un bouton qui vous permet d'envoyer les commandes que vous avez entrées. **Cliquer sur le bouton peut aussi être remplacé par la touche Entrée.**

## II - Les commandes

Nous avons défini plusieurs commandes dans la RFC que nous avons suivie, nous allons donc les rappeler juste ici :

- `/help` : permet d'afficher toutes les commandes existantes avec une courte description de leur fonctionnalité.
- `/signup <username>` : permet de vous inscrire en spécifiant le nom que vous voulez utiliser en paramètre.
- `/msg <message>` ou `<message>` : permet d'envoyer un message à tous les utilisateurs connectés.
- `/msgpv <username> <message>` : permet d'envoyer un message à un utilisateur en particulier en passant son nom en paramètre suivis du message à envoyer.  
*Pour pouvoir envoyer un message privé il vous faut avoir créé un channel avec cet utilisateur.*
- `/afk` : l'utilisateur reçoit toujours les messages des autres utilisateurs mais est "inactif" ce qui engendre qu'il ne peut pas entrer de commandes tant qu'il ne s'est pas remis "actif".
- `/btk` : l'utilisateur repasse au mode "actif", ce qui lui permet d'avoir à nouveau la possibilité d'envoyer des messages.
- `/users` : donne la liste de tous les utilisateurs connectés.
- `/rename <username>` : permet à l'utilisateur de se renommer par le username passé en paramètre.
- `/ping <username>` : permet de notifier un utilisateur.
- `/channel <username>` : permet de créer un channel avec un utilisateur. Ce channel permettra d'envoyer des messages privés entre les deux utilisateurs.
- `/acceptchannel <username>` : permet de répondre à la demande de création d'un channel provenant d'un utilisateur en l'acceptant.
- `/declinechannel <username>` : permet de répondre à la demande de création d'un channel provenant d'un utilisateur en la refusant.
- `/sharefile <username> <filepath>` : permet de partager un fichier à un autre utilisateur en lui spécifiant le nom du fichier.
- `/acceptfile <username> <namefile>` : permet de répondre à l'envoi d'un fichier provenant d'un utilisateur en l'acceptant.
- `/declinefile <username> <namefile>` : permet de répondre à l'envoi d'un fichier provenant d'un utilisateur en la refusant.
- `/exit` : permet de quitter l'application.



### *III - Commencement du tchat*

Pour débiter, vous ne pourrez rentrer que 2 commandes dans le champ de saisie.

- Soit /help pour voir la liste des commandes.
- Soit /signup pour vous inscrire dans le chat.

Une fois que vous vous êtes inscrits, vous avez accès à toutes les commandes citées au dessus.

Une fois que vous voulez fermer l'application, il vous suffit d'entrer la commande /exit ou fermer la fenêtre, et l'application se fermera.