

Documentation d'installation

Une vidéo d'appui à cette documentation et comprenant une démonstration est disponible au lien suivant : [Vidéo](#)

1. Table des matières :

1.	Table des matières :-----	1
2.	Télécharger les codes sources :-----	2
3.	Les dépendances de langages :-----	2
a)	Python-----	2
b)	C et C++ -----	2
c)	JAVA-----	2
4.	Les environnements virtuels :-----	3
5.	Les dépendances Python :-----	3
6.	Démarrages des composantes du projet :-----	4
a)	Le serveur maître :-----	4
d)	Le serveur esclave :-----	4
e)	Le client :-----	5

2. Télécharger les codes sources :

Les composantes de cette SAE sont toutes disponibles dans le repository GitHub R-3.09/SAE-3.02 disponible à l'adresse suivante : [GitHub](#).

Les codes sources à récupérer sont ;

- client vfinal.py
- serveur esclave vfinal.py
- serveur maitre vfinal.py

À noter que le repositionner est paramétré en visibilité privée, seul le compte 'fdrouhin-uha' est accepté en tant que collaborateur du Git. Il faut donc bien vérifier le statut de connexion au compte afin de bien avoir accès au Git.

3. Les dépendances de langages :

Ce projet est entièrement rédigé en python 3.12

Le serveur esclave est capable de s'adapter en fonction des langages installés sur son environnement d'exécution, les langages supportés sont :

a) Python

- Linux : 'sudo apt install python3 | sudo apt install pip3'
- Windows : [Python](#)

b) C et C++

- Linux : 'sudo apt install build-essential)
- Windows : [MinGW](#)

c) JAVA

- Linux : 'sudo apt install default-jdk'
- Windows : [JDK](#)

À noter que le JRE de Java seul ne suffit pas. Le serveur esclave à besoin du JDK de java pour compiler les programmes. En revanche, pas besoin d'installer de JRE, car celui-ci vient avec l'installation du JDK sur Windows et Linux.

Une fois tous les langages installés, il faut bien sûr penser à les ajouter au PATH du système d'exploitation, auxquels cas, le serveur ne sera pas en mesure de les utiliser.

4. Les environnements virtuels :

Les environnements virtuels en python sont conseillés mais non-obligatoire. Ils permettent de pouvoir télécharger des bibliothèques externes à python de façon isolée, et non directement sur la machine hôte. Cette pratique prévient des conflits entre les bibliothèques, permet une meilleure gestion et contrôle des ressources.

Le module venv est installé parmi les modules par défaut de python et permet d'initialiser des environnements virtuels simplement.

Méthode pour initialiser un environnement virtuel via un terminal :

sur Linux :

- `python3 -m venv /chemin/d'accès/venv` #création du venv
- `cd /chemin/d'accès/venv` #navigation vers le dossier du venv
- `bin/activate` #activation du venv

sur Windows :

- `python -m venv C:\chemin\vers\venv` #création du venv
- `cd C:\chemin\vers\venv` #navigation vers le dossier du venv
- `Scripts\activate.bat` #activation du venv

Une fois le venv activé, le prompt du terminal doit avoir mentionné '(venv)'. À noter que celui-ci est actif uniquement dans les terminaux où vous l'avez activé.

5. Les dépendances Python :

Seul les bibliothèques PyQt6 et psutil sont nécessaires au bon fonctionnement du, respectivement client et le serveur esclave. Le serveur maître ne nécessite d'aucun module externe.

sur Linux :

- `pip3 install --upgrade pip` #mise à jour du module pip
- `pip3 install PyQt6` #installation du module PyQt6
- `pip3 install psutil` #installation du module psutil
- `pip3 list` #vérification avec la liste des modules installés

sur Windows :

- `pip install --upgrade pip` #mise à jour du module pip
- `pip install PyQt6` #installation du module PyQt6
- `pip install psutil` #installation du module psutil
- `pip list` #vérification avec la liste des modules installés

À noter qu'à l'heure de la rédaction de cette documentation, PyQt6 n'est pas compatible avec la version 3.13 de python.

6. Démarrages des composantes du projet :

a) Le serveur maître :

Celui-ci est le cœur du projet, il doit impérativement être proprement initialisé pour que l'ensemble fonctionne.

Le serveur maître ne nécessite d'aucune bibliothèque externe à python.

Sur Linux :

- `python3 "/chemin/vers/serveur maitre vfinal.py"` #exécution du script

sur Windows :

- `python -m venv C:\chemin\vers\serveur maitre vfinal.py` #exécution du script

Une fois le script lancé, une boîte de dialogue dédiée à la configuration du serveur se lance. Celle-ci vous demandera de renseigner :

- l'ip du serveur, qui peut être celle de votre réseau local pour une utilisation étendue. Ou loopback (127.0.0.1) pour une utilisation locale.
- Le port de connexion pour les serveurs esclaves,
- Le nombre de serveur esclave maximal connecté au serveur
- Le port de connexion des clients
- Le nombre de client maximal connecté au serveur

Une fois le serveur maître initialisé, bien vérifié que les ports attribués aux clients et aux serveurs sont bien ouvert dans votre pare-feu pour la règle TCP-IP.

'/help' dans le terminal' fournira une liste d'outil utile à la gestion du serveur.

d) Le serveur esclave :

Le serveur esclave nécessite psutil comme module externe à python. Il doit donc être exécuté sur un environnement virtuel, ou non, comprenant ce module python.

À noter que le script peut être exécuté autant de fois que de serveur esclave sont nécessaire dans la limite de la configuration du serveur maître.

Sur Linux :

- `python3 "/chemin/vers/serveur esclave vfinal.py"` #exécution du script

sur Windows :

- `python -m venv C:\chemin\vers\serveur esclave vfinal.py` #exécution du script

Une fois le script lancé, une boîte de dialogue se lancera également à la configuration du serveur esclave, demandant de renseigner les informations du serveur maître.

'/help' dans le terminal' fournira une liste d'outil utile à la gestion du serveur.

e) Le client :

Le client nécessite PyQt6 comme module externe à python. Il doit donc être exécuté sur un environnement virtuel, ou non, comprenant ce module python.

À noter que le script peut être exécuté autant de fois que de serveur esclave sont nécessaire dans la limite de la configuration du serveur maître.

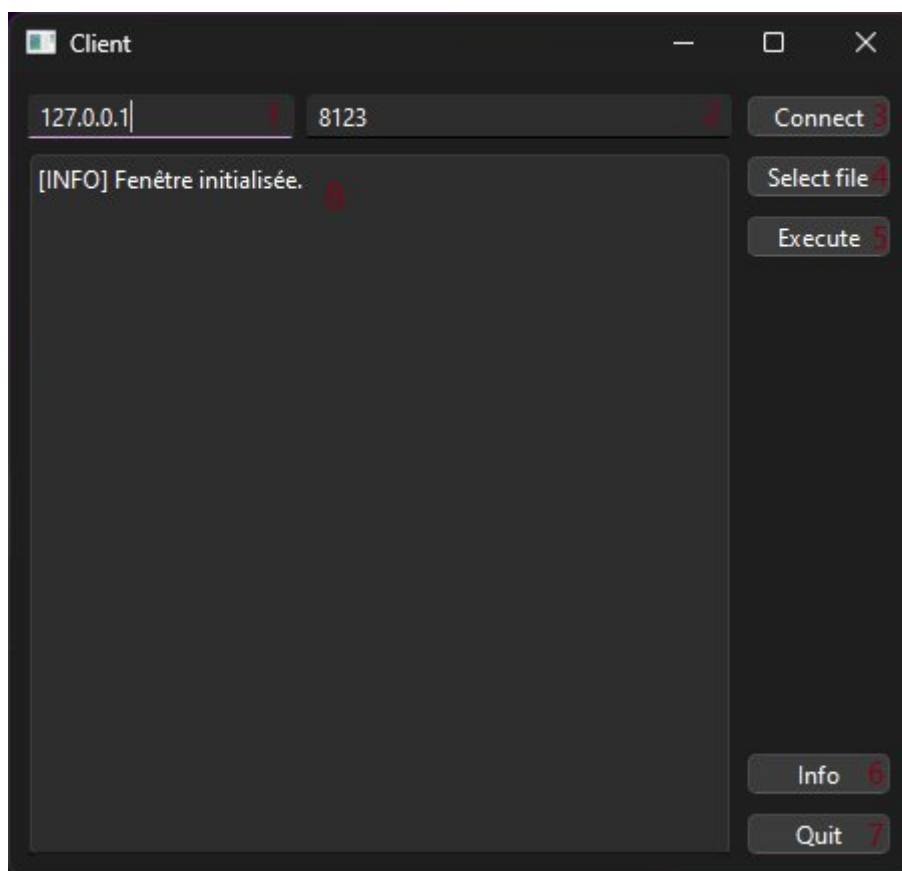
Sur Linux :

- `python3 "/chemin/vers/client vfinal.py"` #exécution du script

sur Windows :

- `python -m venv C:\chemin\vers\client vfinal.py` #exécution du script

La fenêtre suivante s'ouvrira :



- 1 : Ip du serveur maître à renseigner
- 2 : Port client du serveur maître à renseigner
- 3 : Bouton de connexion au serveur maître
- 4 : Bouton de sélection de fichier à envoyer
- 5 : Bouton d'envoi au serveur
- 6 : Affichage du statut de connexion dans la console
- 7 : Fermeture du client
- 8 : Console du client