**Rapport de la SAE 302**

Dans ce document, je vais vous expliciter tout ce que j’ai réalisé pour mener à bien ce projet. C’est le résumé des critères que j’ai respectés, réalisés, ou non …

Je rentre plus dans le détail du projet dans la documentation d’installation et la documentation de mes codes.

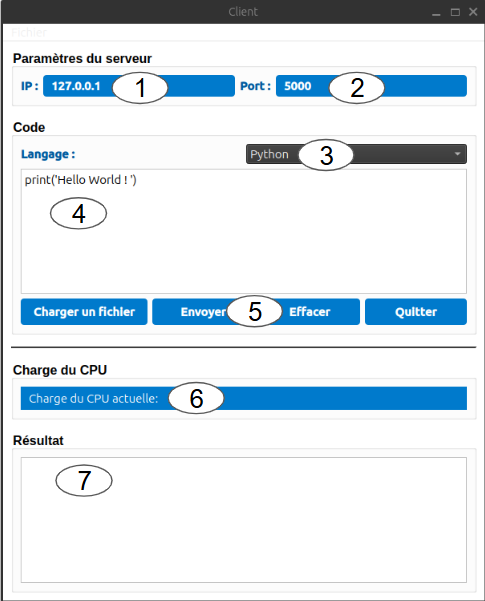
Cette documentation se compose en 3 parties:

[**→ Le(s) Client(s)**](#Client)

[**→ Le Serveur Maître**](#Serveur)

[**→ Le Serveur Esclave**](#ServeurEsclave)

**Partie Client**



1. **Adresse IP** du Serveur utilisé (ici, le Serveur Maître)
2. **Port du Serveur** utilisé (5000 → Serveur Maître)
3. Choix du **langage à compiler ou interpreter** (python, java ou C)
4. **Zone d’affichage du code** à compiler ou interpréter
5. Tous les boutons pour -**Charger un fichier** (importer depuis l’ordinateur)

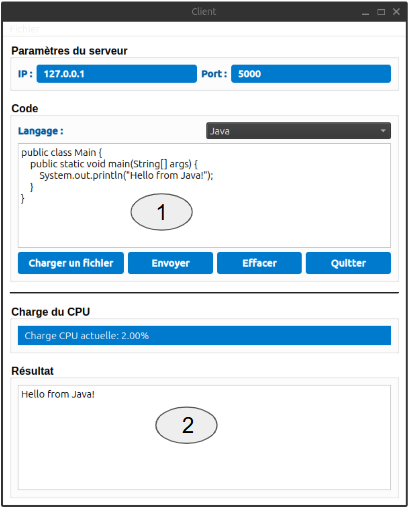
-**Envoyer au Serveur le fichier** qu’on vient de choisir

-**Effacer cette zone** pour choisir un autre fichier

-**Quitter proprement** l’interface graphique

1. Zone d’affichage de la **charge CPU** que le Serveur Maître renvoi
2. La toute dernière zone qui **affiche le Résultat** du fichier sélectionné précédemment. C’est la partie la plus importante car c’est l’objectif principal de cette interface graphique.

Mon Client est très simple d’utilisation et assez intuitif.



Voici un exemple concret de l’envoi d’un fichier en java.

on peut voir que le contenu du fichier est affiché dans la zone 1 et le résulat dans la zone 2.

Tout mes clients ont le même code source donc je peux en ajouter autant que je souhaite.

**Partie Serveurs**

**Serveur.py**



Voici ce que renvoie le cmd de mon code Serveur.py quand je le lance.

On s’aperçoit que les données essentielles sont précisées.

Je lance mon script python Serveur.py dans un CMD (windows ou Linux) avec la commande suivante :

windows → python Serveur.py

Linux → python3 Serveur.py

Le serveur maître se lance en indiquant bien qu’il écoute sur l’adresse 127.0.0.1 et sur le port n°5000.

Vous allez comprendre par la suite pourquoi j’ai choisi le port n°5000.

En réalité, lorsque je teste le projet à partir d’une VM, il faut penser à modifier l’adresse IP sur l’interface graphique par celle de la VM qui héberge le Serveur Maître.

On s’aperçoit aussi que le Serveur indique “MAX\_LOCAL\_TASKS=2”

→ Cela veut simplement dire que si le serveur maître reçoit plus de 2 programmes à interpréter ou compiler en même temps, il va transmettre les prochaines tâches au serveur esclave (ServeurEsclave.py).

Mon Serveur est donc capable d’éxecuter des codes en python, en java et en C en hébergeant plusieurs clients.

**ServeurEsclave.py**



De même pour le Serveur Esclave (ServeurEsclave.py), lorsque je le lance le script, il affiche l’adresse IP sur laquelle il écoute ainsi que le son port.

Comme pour le Serveur Maître, quand je test le projet à partir d’une VM, il faut penser à modifier l’adresse IP sur l’interface graphique par celle de la VM qui héberge le Serveur Maître.

Le Serveur maître écoute donc sur le port 5000 et le Serveur Esclave sur le port 6000.

On va voir par la suite que lorsque les clients envoi plus de 2 programmes en même temps, le serveur maître va faire passer le reste des programmes sur le port 6000 (Serveur Esclave).

C’est donc le Serveur Esclave qui prend en charge tous les clients qui dépassent le MAX\_LOCAL\_TASKS.