**ROBIN**

**Mathias**

**RT22 cyber FA**

**Document de réponse** **SAE3.02**

# Sommaire

[**I. Sommaire**](#_3sf8lre1458l) **1**

[**II. Présentation**](#_pbs4irqcfy5l) **1**

[**III. Réponse au cahier des charges**](#_bnvcbsxkocnr) **1**

[1. Coder un serveur](#_ct45vkkr3adt) 1

[2. Coder un client](#_zef8tj6qhzvl) 2

[**IV. Conclusion**](#_67wmapt587kv) **2**

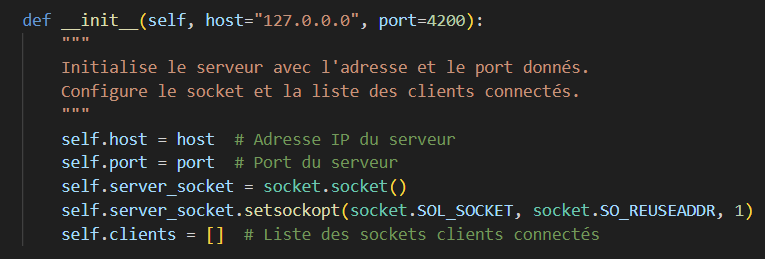
# Présentation

Lors de la réalisation du projet SAE3.02 nous avons dû réaliser un programme en python dans le but de créer un serveur qui exécute les programmes (en C, C++, Java et python) d’un client, le serveur doit pouvoir recevoir le programme, le compilé si besoin, l'exécuter et le renvoyer au client afin de l’afficher. Le client quant à lui dispose d’une interface graphique afin d’envoyer ces programmes et d’en afficher la sortie. Ce document fait office de de document de réponse au cahier des charges du sujet simple du projet SAE3.02.

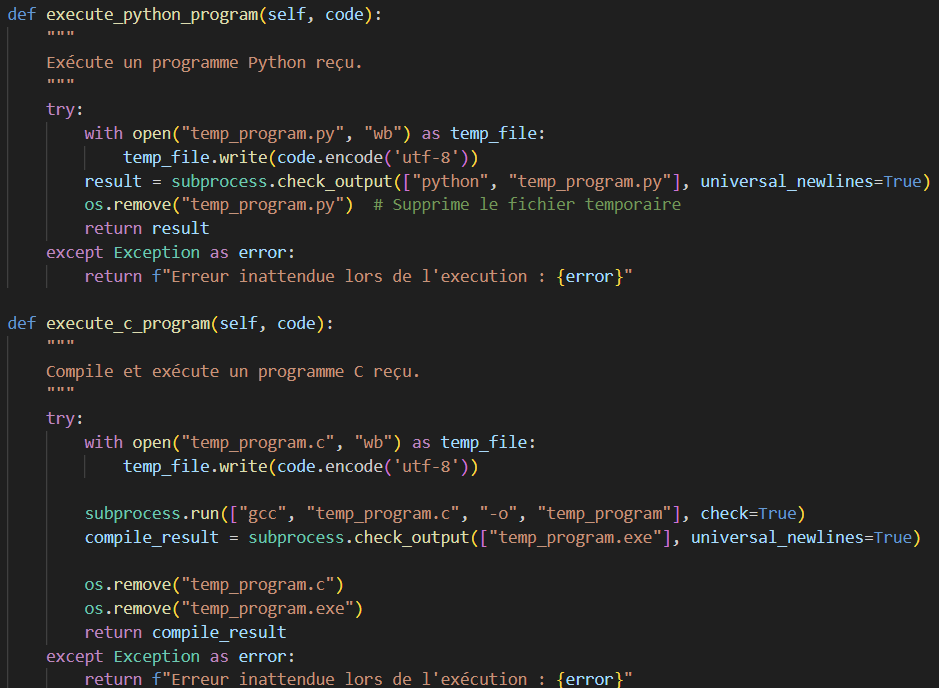
# Réponse au cahier des charges

## Coder un serveur

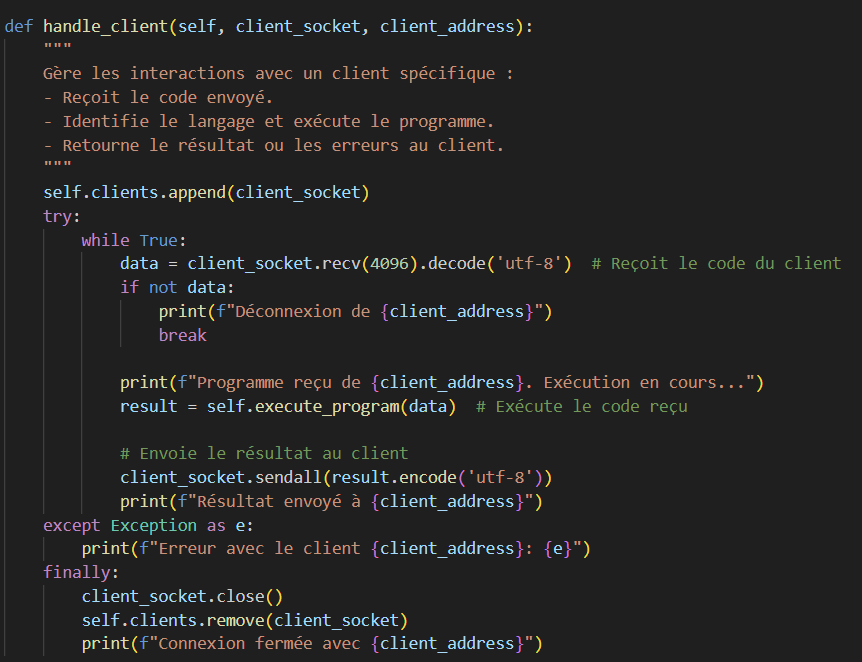
* **Connexion client-serveur :** 
  1. Le serveur reçoit d’abord une requête d’un client



* 1. Le serveur accepte le programme envoyé par le client, le compile (si nécessaire) et l'exécute.

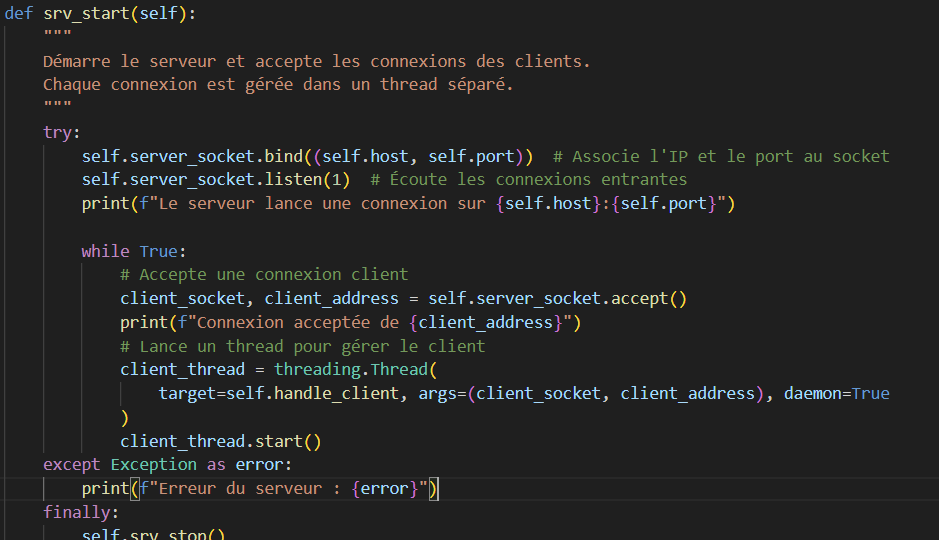


* 1. Les résultats ou les erreurs sont renvoyés au client.



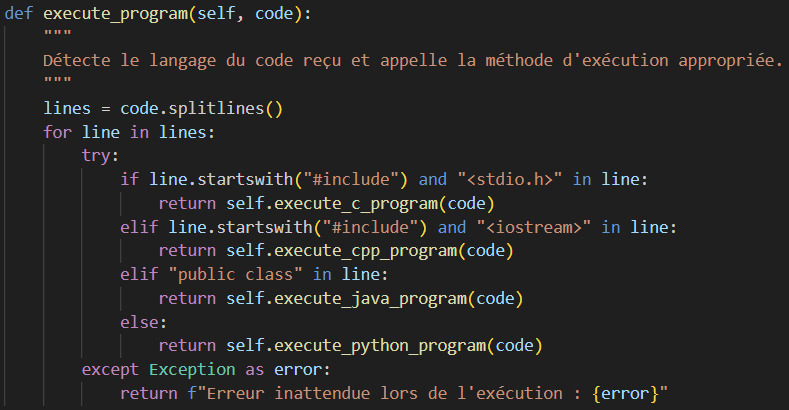
* **Gestion de multiples clients** :

1. Le serveur gère plusieurs clients simultanément grâce à des threads, cependant le programme ne détecte pas lorsqu’un client est déjà connecté au serveur ou non. C’est une des fonctionnalités que je n'ai pas réussi à implémenter.



* **Support de plusieurs langages** :

1. Le serveur supporte Python, C, C++ et Java, détectant automatiquement le langage du programme reçu.

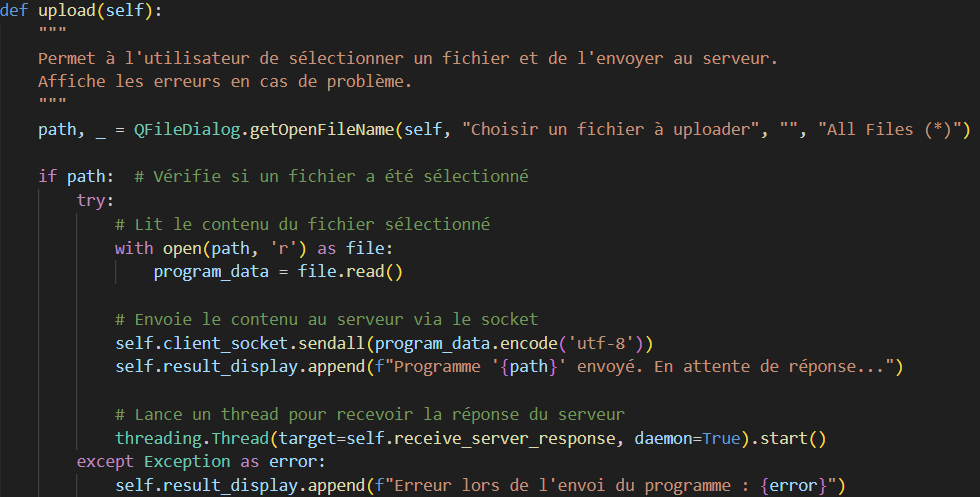


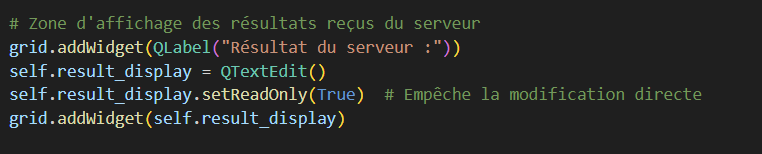
## Coder un client

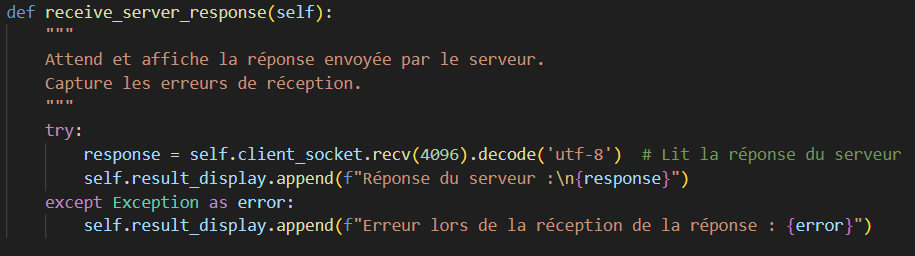
1. Le client se connecte à un serveur via une interface graphique développée avec PyQt, permettant de spécifier une adresse IP et un port (valeurs par défaut fournies).



1. L'utilisateur peut sélectionner un fichier à envoyer.



1. Les résultats ou les erreurs sont affichés dans une zone de texte dédiée.



1. Lorsque le serveur rejette le programme, l'utilisateur peut renvoyer un autre programme directement.

# Conclusion

Ce projet m’a pris environ 30 h pour tout réaliser, du programme au document à rendre, comme j’ai choisi le sujet simple je pensais pouvoir terminer ce projet plus rapidement cependant j’ai eu des difficultés avec la compilation et l'exécution des programme en C, C++ et java, c’était quelque chose qu’on avait jamais fait auparavant, c’est ce qui m'a pris le plus de temps à debug.

En ce qui concerne les améliorations du code potentiel j’ai pensé au côté cyber comme par exemple l’ajout d’un login à la page de connexion du client, on aurait pu penser aussi à l’implémentation d’une connexion chiffrée entre le client et le serveur.