



Documentation d'Installation et de Configuration

Cette documentation explique comment installer et configurer le projet depuis le dépôt GitHub, ainsi que les prérequis nécessaires pour le faire fonctionner.

1. Pré-requis

Avant de commencer, assurez-vous d'avoir les éléments suivants :

- **Python**.
- **Git** pour cloner le dépôt.
- Les modules Python nécessaires :
 - PyQt6
 - psutil
- Un système d'exploitation pris en charge :
 - **Windows** 10 ou supérieur.
 - **Linux** (Debian, Ubuntu, etc.).
 - Mac (**ATTENTION** : Aucun test n'a été fait sur cette OS, je ne suis donc responsable d'aucun problème rencontré.)

2. Récupération du code source (deux méthodes)

Le faire en ligne de commande :

1. Ouvrez un terminal ou une invite de commande.
2. Clonez le dépôt GitHub avec la commande suivante :

```
git clone https://github.com/Sylverthorn/GIT-R309-SAE302.git
```

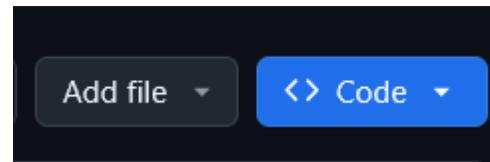
3. Accédez au dossier du projet :

```
cd GIT-R309-SAE302/SAE_3.02/
```

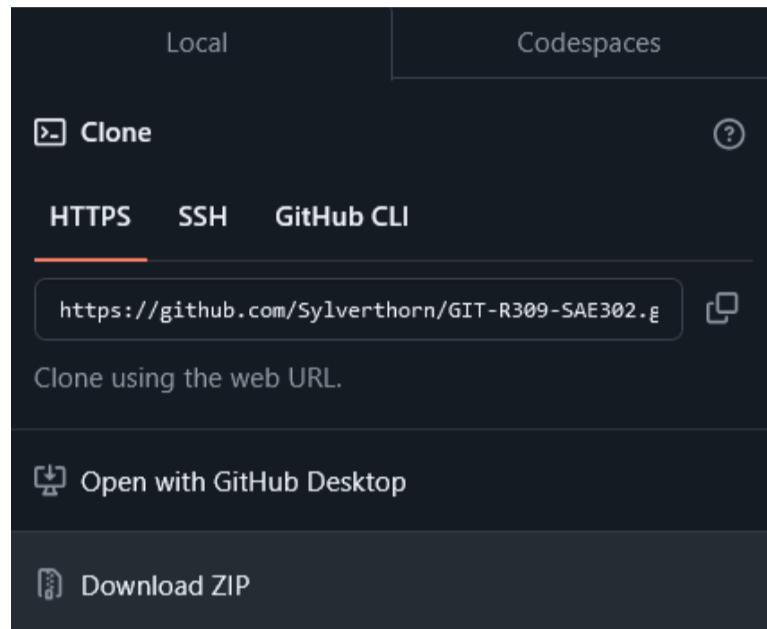
Télécharger sur le web :

- Se rendre sur : ➡ <https://github.com/Sylverthorn/GIT-R309-SAE302.git>

- cliquer sur `<> Code` :



- Une fenêtre s'affiche, puis cliquer sur **Download Zip** :



3. Installation des dépendances

1. Créez un environnement virtuel Python (optionnel mais recommandé) :

```
python -m venv env
source env/bin/activate      # Pour Linux/Mac
env\\Scripts\\activate        # Pour Windows
```

2. Installez les dépendances nécessaires:

```
pip install PyQt5
pip install psutil
```

3. Ou alors vous pouvez aussi les installer en brute directement sans passer par un VENV :

```
pip install PyQt5
pip install psutil
```

4. Configuration et Exécution

ATTENTION :

Si vous avez décidé d'utiliser un venv n'oubliez pas de lancer les scripts dans ce même venv
(rappel ci dessous des commandes pour être dans le venv) :

```
source env/bin/activate      # Pour Linux/Mac
env\\Scripts\\activate        # Pour Windows
```

Une dernière chose :

Tous les scripts à lancer doivent être lancés impérativement dans le dossier **SAE_3.02/**

```
cd GIT-R309-SAE302/SAE_3.02/  
ou  
cd SAE_3.02/ #si vous êtes déjà dans GIT-R309-SAE302/
```

Serveur Maître

1. Lancez le script `interface_server.py` pour démarrer le serveur maître avec une interface graphique :

```
python interface_server.py
```

2. Configurez les ports d'écoute pour les clients et serveurs secondaires via l'interface graphique.
3. Activez ou désactivez le lancement automatique des serveurs secondaires selon vos besoins.
 - a. Si lancement automatique Activé, configurer les serveurs secondaires sur cette interface

Serveurs Secondaires

- **Option 1** : Si le lancement automatique est activé, les serveurs secondaires seront démarrés automatiquement via `subprocess`.
- **Option 2** : Si le lancement manuel est préféré :
 - Accédez au script `serversecondaire.py`.
 - Lancez-le sur les machines souhaitées :

En remplissant bien les différents argument demandé, comme ci-dessous :

```
python serversecondaire.py <ip_maitre> <port_maitre> <nb_tache_max> <cpu_max>
```

Client

1. Lancez l'interface client avec le script `interface_client.py` :

```
python interface_client.py
```

2. Configurez l'adresse IP et le port du serveur maître.
3. Sélectionnez un fichier dans le dossier `test/` ou un de vos propres fichiers à exécuter, modifiez-le si nécessaire, puis envoyez-le au serveur maître via l'interface.

5. Arborescence du Projet

```
SAE_3.02/  
|  
|   __ client.py           # Script de base du client  
|   __ interface_client.py # Interface graphique du client  
|   __ interface_server.py # Interface graphique du serveur maître  
|   __ server_second.py    # Script des serveurs secondaires  
|   __ server.py           # Script de base du serveur maître  
|   __ style.qss            # Fichier de style pour les interfaces  
|  
|   __ execution/          # Dossier pour gérer les fichiers en cours d'exécution  
|   __ test/                # Dossier contenant des fichiers de test  
|   README.md              # Fichier de documentation
```

Lien vers le projet GitHub : [SAE_3.02](#)