

Fiche d'installation pour le projet SAE3.02

Prérequis

1. Une machine virtuelle Linux (Ubuntu ou similaire) avec accès à internet.
2. Les outils suivants installés sur la machine :
 - Python 3
 - GCC (pour les programmes C et C++)
 - G++ (pour les programmes C++)
 - Java Development Kit (JDK)
 - PyQt6 (pour l'interface graphique du client)
3. Les fichiers du projet disponibles sur GitHub : <https://github.com/Hamedalabri/SAE3.02>.

Note : Si vous cliquez sur le lien et recevez une erreur 404, essayez de copier le lien et de le coller dans un autre navigateur web.

Étapes d'installation

1. Téléchargement du projet

- Téléchargez le fichier ZIP du projet depuis le dépôt GitHub :
<https://github.com/Hamedalabri/SAE3.02>.
- Déplacez-vous dans le répertoire Downloads (ou l'endroit où le fichier a été téléchargé) :
 - `cd ~/Downloads`
- Extrayez le contenu du fichier ZIP :
 - `unzip SAE3.02-main.zip`

2. Vérification et installation de Python

- Vérifiez si Python et Python 3 sont installés :
 - `python --version`
 - `python3 --version`
- Si Python ou Python 3 n'est pas installé, exécutez la commande suivante :
 - `sudo apt update`
 - `sudo apt install python3 python3-pip -y`

3. Installation des dépendances

3.1. Bibliothèques associées à Python

- Installez le module Python pour les environnements virtuels si nécessaire :

- `sudo apt install python3-venv -y`
- Créez un environnement virtuel pour installer les bibliothèques nécessaires :
 - `python3 -m venv env`
 - `source env/bin/activate`
- Installez PyQt6 dans l'environnement virtuel :
 - `pip install PyQt6`

3.2. Compilateurs pour les autres langages

- Installez GCC et G++ pour C/C++ :
 - `sudo apt install gcc g++ -y`
- Installez le JDK pour Java :
 - `sudo apt install default-jdk -y`

4. Configuration du serveur

1. Naviguez dans le dossier contenant les fichiers serveur :
 1. `cd ~/Downloads/SAE3.02-main`
2. Lancez le serveur en spécifiant le port souhaité (par défaut : 12345) :
 1. `python3 Server.py <port>`

5. Lancement du client

1. Rendez-vous sur [GitHub](https://github.com) et téléchargez le fichier ZIP du projet.
2. Extrayez le contenu du fichier ZIP dans un dossier de votre choix.
3. Ouvrez Visual Studio Code (VS Code) sous Windows.
4. Naviguez dans le répertoire contenant les fichiers client en utilisant l'explorateur de fichiers de VS Code ou ouvrez le dossier :
 - Cliquez sur "File" > "Open Folder" et sélectionnez le dossier du projet.
5. Vérifiez si Python et Python 3 sont installés :
 - `python --version`
 - `python3 --version`
6. Si Python ou Python 3 n'est pas installé, exécutez les commandes suivantes dans un terminal pour les installer :
 - `sudo apt update`
 - `sudo apt install python3 python3-pip -y`
7. Assurez-vous que Python est configuré dans VS Code.
8. Lancez le client en exécutant le fichier `Client.py` directement dans le terminal intégré ou en utilisant l'option "Run" de VS Code.

6. Envoi et exécution de programmes

1. Avant de vous connecter au serveur, exécutez la commande suivante pour connaître l'adresse IP de votre serveur :
 - `ip a`

Notez l'adresse IP associée à votre interface réseau.

2. Une fois le client connecté au serveur, utilisez l'interface graphique pour :
 - Spécifier l'adresse IP et le port du serveur.
 - Charger un fichier contenant le programme à exécuter.
 - Envoyer le programme au serveur pour compilation et exécution.
3. Le serveur détectera automatiquement le langage, compilera et exécutera le programme, puis retournera les résultats.

7. Tests et validation

- Utilisez les exemples fournis dans le projet pour tester chaque fonctionnalité :
 - **Python** : python.py
 - **C** : LangC.c
 - **C++** : C++.cpp
 - **Java** : Langjava.java

8. Dépannage

- **Problème de connexion** : Assurez-vous que le serveur est actif et que le port est correctement configuré.
- **Erreur de compilation ou d'exécution** : Vérifiez que le fichier source est valide et que les outils requis sont installés.

9. Arrêt du serveur

- Pour arrêter le serveur, utilisez **Ctrl+C** dans le terminal.

10. Lancer plusieurs serveurs et clients

1. **Lancer un autre serveur sur un port différent :**
 - Dans un autre terminal, naviguez vers le dossier contenant les fichiers serveur :
 - `cd ~/Downloads/SAE3.02-main`
 - Lancez un deuxième serveur en spécifiant un port différent (par exemple, 12346) :
 - `python3 Server.py 12346`
2. **Lancer un autre client :**
 - Ouvrez un nouveau terminal dans Visual Studio Code.
 - Naviguez vers le répertoire contenant le fichier Client.py si ce n'est pas déjà fait.
 - Tapez la commande suivante pour lancer un deuxième client :
 - `python Client.py`
3. **Lancer un autre serveur sur un port différent :**

- Dans un autre terminal, naviguez vers le dossier contenant les fichiers serveur :
 - `cd ~/Downloads/SAE3.02-main`
- Lancez un deuxième serveur en spécifiant un port différent (par exemple, 12346)
:
 - `python3 Server.py 12346`

Conclusion

En suivant ces étapes, vous aurez un environnement fonctionnel pour le projet SAE3.02, capable de compiler et d'exécuter des programmes écrits en Python, C, C++, et Java.