

Pour mettre votre projet "Synchro Qt Made" sur GitHub, vous allez utiliser Git (un système de contrôle de version) localement, puis le lier à un dépôt distant sur GitHub.

Prérequis :

1. **Git installé :** Assurez-vous que Git est installé sur votre machine. Vous pouvez le vérifier en ouvrant un terminal et en tapant `git --version`. Si ce n'est pas le cas, suivez les instructions sur git-scm.com.
2. **Compte GitHub :** Vous devez avoir un compte GitHub. Si vous n'en avez pas, créez-en un sur github.com.

Étapes :

Étape 1 : Initialiser un Dépôt Git Local

1. **Ouvrez votre terminal ou invite de commande.**
2. **Navigation jusqu'au répertoire racine de votre projet.** C'est le dossier qui contient `Synchro_qt_made`, `Synchro_backend`, `api_client.py`, etc.

```
cd ~/projects/Synchro_qt_made/.. # Ou le chemin réel vers le dossier parent
# Par exemple, si Synchro_qt_made est dans ~/projects/, faites:
# cd ~/projects/
```

Assurez-vous d'être dans le répertoire qui contient `Synchro_qt_made` et `Synchro_backend`.

3. **Initialisez un nouveau dépôt Git :**
`git init`

Cela crée un sous-répertoire `.git` caché, qui contient tous les fichiers nécessaires au suivi des versions.

Étape 2 : Créer un Fichier `.gitignore`

Il est crucial de ne pas stocker certains fichiers sur GitHub (comme les logs, les fichiers de configuration locaux, les environnements virtuels).

1. **Créez un fichier nommé `.gitignore`** (sans nom, juste l'extension) à la racine de votre dépôt Git (le même répertoire où vous avez fait `git init`).
2. **Ajoutez les entrées suivantes à ce fichier :**

```
# Environnements virtuels Python
.venv/
venv/
env/
*.pyc
__pycache__/
```

```
# Fichiers de log de l'application
.synchro/
*.log
logs/
```

```
# Fichiers de configuration spécifiques à l'utilisateur ou à l'environnement
*.json # Si vous avez des fichiers JSON qui ne doivent pas être publics (ex:
volume_config.json)
# Si vos configs sont génériques et peuvent être partagées, ne mettez pas *.json
# ou soyez plus spécifique comme:
# .synchro/volume_config.json
# .synchro/configs/*.json # Si vous ne voulez pas partager les configs sauvegardées
```

Adaptez les lignes .json en fonction de ce que vous voulez partager ou non. Si volume_config.json contient des chemins spécifiques à votre machine, il est préférable de l'ignorer.

Étape 3 : Ajouter les Fichiers au Dépôt Git

1. **Ajoutez tous les fichiers de votre projet (sauf ceux ignorés par .gitignore) :**
git add .

Le . signifie "tous les fichiers et répertoires dans le répertoire courant".

2. **Vérifiez le statut pour voir ce qui sera inclus :**
git status

Vous devriez voir tous vos fichiers de code listés en vert, et les fichiers ignorés ne devraient pas apparaître.

Étape 4 : Effectuer le Premier Commit

1. **Enregistrez les changements dans votre dépôt local :**
git commit -m "Initial commit: First version of Synchro Qt Made"

Le message après -m doit être descriptif de ce que vous avez ajouté ou modifié.

Étape 5 : Créer un Dépôt sur GitHub

1. **Connectez-vous à votre compte GitHub.**
2. **Cliquez sur le bouton "New repository"** (ou le signe + en haut à droite, puis "New repository").
3. **Remplissez les informations :**
 - **Repository name :** Choisissez un nom pour votre projet (ex: Synchro-Qt-Made).
 - **Description (Optional) :** Une courte description.
 - **Public ou Private :** Choisissez "Public" si vous voulez que tout le monde puisse le voir, ou "Private" si vous voulez le garder pour vous ou des collaborateurs spécifiques.
 - **NE cochez PAS "Add a README file", "Add .gitignore", ou "Choose a license".**
Vous les avez déjà créés localement.
4. **Cliquez sur "Create repository".**

Étape 6 : Lier le Dépôt Local au Dépôt GitHub

Après avoir créé le dépôt sur GitHub, vous verrez une page avec des instructions. Cherchez la section "...or push an existing repository from the command line".

1. **Copiez la première ligne fournie par GitHub** (elle ressemblera à git remote add origin

`https://github.com/votre_nom_utilisateur/votre_repo.git`).

- Collez-la dans votre terminal et appuyez sur Entrée.
- Cette commande lie votre dépôt local à votre dépôt distant sur GitHub.

Étape 7 : Pousser les Fichiers vers GitHub

1. Poussez vos commits locaux vers GitHub :

`git push -u origin master`

Ou 'main' si votre branche par défaut est 'main' (GitHub a changé le nom par défaut)

`git push -u origin main`

- `-u origin master` (ou `main`) définit la branche `master/main` de votre dépôt distant (`origin`) comme la branche par défaut pour les futures commandes `git push` et `git pull`.
- Il vous sera demandé de vous connecter à GitHub (nom d'utilisateur et mot de passe ou Personal Access Token si vous avez l'authentification à deux facteurs).

Félicitations ! Votre projet est maintenant sur GitHub. Vous pouvez visiter l'URL de votre dépôt pour le voir.

N'hésitez pas si vous avez d'autres questions ou si vous rencontrez des difficultés lors de ces étapes !