

Utiliser Git avec Windows et VSC

Dans cet article, tu vas découvrir les étapes nécessaires pour installer git sous Windows et l'utiliser dans Visual Studio code sans ligne de commande.

5TTR

6TTR

6TQ



Découverte

Installation et configuration de Git sur Windows

1. Installer Git

- Télécharge Git depuis : <https://git-scm.com/download/win>
- Lance l'installation et garde les réglages par défaut.
- Vérifie l'installation en ouvrant un terminal (PowerShell ou Git Bash) :

```
1 | git --version
```

Si tout va bien, tu verras quelque chose comme :

```
git version 2.45.0.windows.1
```

2. Configurer Git (une seule fois)

Dans le terminal, configure ton nom d'utilisateur et ton email (ceux utilisés sur GitHub) :

```
1 | git config --global user.name "Ton Nom"
2 | git config --global user.email "ton_email@exemple.com"
```

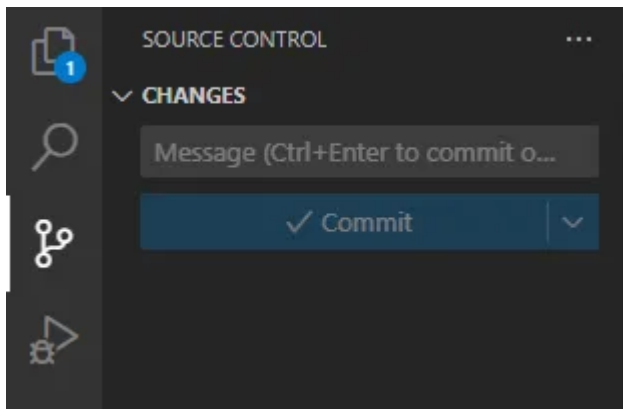
Tu peux vérifier ta configuration avec :

```
1 | git config --global --list
```

Utiliser Git avec Visual Studio Code

Installation du plugin Git dans VS Code

- Ouvre Visual Studio Code.
- Normalement, VS Code possède déjà Git intégré. Pour le vérifier, ouvre un dossier dans VS Code :
 - Clique sur l'icône « Source Control » (en forme de branche) à gauche.



Tu devrais voir apparaître un bouton "Initialize Repository".

Première utilisation : créer un nouveau dépôt (repository)

- Clique sur « Initialize Repository » ou tape dans le terminal :

```
1 | git init
```

Ceci crée un dossier caché `.git` qui gère ton suivi des versions.

Workflow de base avec Git et GitHub

Voici l'essentiel à maîtriser pour bien utiliser Git :

1. Faire des modifications et vérifier l'état de ton projet

- Ajoute/modifie un fichier dans ton dossier.
- Vérifie l'état de ton projet avec :

```
1 | git status
```

2. Préparer les modifications (« staging »)

Git fonctionne en deux étapes :

1. Préparer les fichiers modifiés (« stage »).
2. Sauvegarder ces modifications (« commit »).

Pour préparer les modifications :

```
1 | git add nom_du_fichier
```

Pour préparer tous les fichiers :

```
1 | git add .
```

3. Enregistrer les modifications (« commit »)

Sauvegarde les modifications dans ton historique avec un message explicatif :

```
1 | git commit -m "Ajoute une nouvelle fonctionnalité"
```

4. Connecter Git à GitHub et envoyer les modifications (« push »)

- Crée un compte sur [GitHub.com](https://github.com).
- Crée un nouveau dépôt (repository) vide sur GitHub (sans README, sans fichiers).
- Copie l'adresse HTTPS du dépôt (ex : https://github.com/ton_pseudo/ton_projet.git).

Ensuite, depuis ton terminal VS Code :

```
1 | git remote add origin https://github.com/ton_pseudo/ton_projet.git
2 | git branch -M main
3 | git push -u origin main
```

Maintenant, ton code est sauvegardé en ligne sur GitHub.

5. Récupérer des modifications depuis GitHub (« pull »)

Si tu modifies ton projet depuis un autre ordinateur ou directement depuis GitHub, récupère ces modifications avec :

```
1 | git pull
```

Bonnes pratiques à retenir

- Fais des commits régulièrement, avec des messages courts mais clairs.
 - Toujours faire un `git pull` avant de commencer à travailler.
 - Ne jamais publier de données sensibles sur GitHub (ex : mots de passe).
-

Tu as maintenant tout en main pour utiliser Git efficacement, même tout seul !