

GIT Windows / GitLab Paris-Saclay

Rédigé par Julien Villemejeane / LEnsE – Institut d'Optique

Version de travail – 11/01/2023 – A compléter

Ressources

- Understand Git : <https://nvie.com/posts/a-successful-git-branching-model/>
- How to : <https://docs.gitlab.com/ee/gitlab-basics/start-using-git.html>

Installation du client GIT pour Windows

Il est possible de récupérer une version du client Git pour Windows à l'adresse suivante : <https://git-scm.com/download/win>

Une version portable (que l'on peut copier dans une clef USB par exemple) est également disponible et permet d'utiliser GIT sans installation préalable sur une machine.

Création d'un projet GITLab Paris-Saclay

A l'aide de votre compte Adonis (compte mail), il est possible de se créer un espace GitLab sur le serveur de l'université Paris-Saclay : <https://gitlab.dsi.universite-paris-saclay.fr/>

Connectez-vous sur cette plateforme.

Il est alors possible de créer un **nouveau projet** en lui donnant un **nom** et en précisant le **type d'accès** : privé (à l'administrateur d'ajouter des utilisateurs), interne (accessible à toute la communauté Paris-Saclay – avec restrictions d'accès possible), publique (accessible à toute personne sans restriction d'accès en lecture).

Utilisation de GIT pour Windows

Création d'un répertoire pour GIT et lancement de GIT

Créer un répertoire pour git (type git_repo) sur votre propre ordinateur.

Lancer **GITBash** (ou git CMD).

Dans la fenêtre qui s'ouvre :

```
> cd git_repo
```

Il est possible de configurer les paramètres de base de GIT

```
> git config --global user.name "Julien Villemejeane"  
> git config --global user.email julien.villemejeane@universite-paris-saclay.fr  
> git config --global init.defaultBranch master
```

Selon la configuration du serveur, il est possible qu'un mot de passe vous soit demandé.

Clonage d'un répertoire existant

La première étape, si un dépôt GitLab ou GitHub existe, est de cloner le répertoire distant dans le répertoire local. Pour cela, il faut utiliser la commande suivante :

```
> git clone <http_address>  
(par exemple : https://github.com/jvillemejeane/SupOpToolboxes.git )
```

Ensuite, il faut se déplacer dans le nouveau répertoire :

```
> cd <git_directory>
```

Informations sur le dépôt GIT

```
> git status // to get status  
> git branch // to display the actual branch  
> git checkout //
```

Information sur les différentes branches

Pour connaître les différentes branches du répertoire :

```
> git branch
```

Changement de branche

Pour changer de branche :

```
> git checkout <branch-name>
```

Vérification d'une nouvelle version sur le serveur - Pull

Pour vérifier qu'une nouvelle version n'a pas été téléversée par un autre utilisateur sur le serveur distant, il faut exécuter la commande :

```
> git checkout
```

Cette commande vérifie la branche en cours d'utilisation. Sinon :

```
> git checkout <branch-name>
```

Si une nouvelle version a été téléversée, il est alors possible de la récupérer par la commande :

```
> git pull
```

Nouveaux fichiers ou répertoires - Add / Commit / Push

Lorsqu'on ajoute des fichiers ou répertoires

```
> git add <file_or_dir_name>
```

> **git status** // to confirm that files or directories are waiting to be add on the server

S'il y a plusieurs fichiers à ajouter : > **git add -A**

Il faut ensuite renseigner la mise à jour

> **git commit -a -m "COMMENT TO DESCRIBE THE INTENTION OF THE COMMIT"**

Enfin, il faut téléverser les fichiers/répertoires vers le serveur depuis la branche en cours

> **git push**

Si la branche a été créée en local mais pas encore téléverser vers le serveur, il faut utiliser la commande suivante :

> **git push -u origin <branch name>**

Création d'une nouvelle branche - Checkout

Pour créer une nouvelle branche locale à partir d'une branche existante :

> **git checkout -b <new branch name> <original branch name>**

Fusion de deux branches - Merge / Merge-request

Lors de la fusion de plusieurs branches ensemble, il est essentiel de ne pas effacer des mises à jour faites par d'autres utilisateurs et de vérifier le contenu de chacune des versions afin de décider quels fichiers (ou partie de fichier) il est intéressant de maintenir.

Cette phase est souvent réalisée par le chef de projet qui va demander l'avis des différents développeurs.

Pour réaliser cette étape, il faut, sous GitLab, faire une demande de fusion (merge request). Pour cela, il faut se rendre sur la page du projet GitLab et se rendre dans la partie "**Merge requests**".

Puis demander une nouvelle requête : **New Merge Request**. Il faut alors sélectionner quelle branche fusionner avec quelle autre et préciser les raisons de la fusion.