Polytech Lille — IS4 — Algorithmique Numérique pour l'Optimisation — Résumé git

---

Les `<expressions entre chevrons>` qui apparaissent dans les commandes doivent être remplacées par leur valeur.

1 Création de votre projet sur le gitlab

On appelle *gitlab* la machine `https://gitlab.univ-lille.fr` qui héberge les *projets*.

Pour accéder au gitlab, il est nécessaire de disposer de son identifiant univ-lille (celui qui se termine par "`.etu`") et de son mot de passe (ne pas confondre avec l'identifiant Polytech). En cas d'oubli du mot de passe : `https://sesame.univ-lille.fr`

Une première version du projet (appelée *projet initial*), nommé `is4-ano-2022`, a été créée sur le gitlab par un enseignant. Les opérations suivantes doivent être exécutées sur le gitlab, une seule fois. Elles créent un projet pour chaque étudiant, par duplication du *projet initial*, et permettent aux enseignants d'y accéder pour interagir avec vous

- Chercher le projet `is4-ano-2022` (`Explore Projects`)
- L'ouvrir et cliquer sur `Fork` (en haut à droite). Votre projet est créé. Les commandes suivantes le configurent.
- Cliquer sur `Settings` (barre latérale) et s'assurer que le projet est privé (`project visibility : private`)
- Cliquer sur `Members` (barre latérale) et ajouter les enseignants suivants avec le statut (`role permission`) `maintainer` : François Boulier, François Lemaire, Kevin Feghoul

2 Création (clone) d'une copie de travail

On appelle *copie de travail* un répertoire situé sur une machine de travail (ordinateur personnel ou machine de TP de Polytech) qui duplique le contenu d'un (de votre) projet. Sur un même ordinateur, on peut disposer de plusieurs copies de travail. À partir du moment où une copie de travail est disponible, il ne devrait plus y avoir aucune modification ou ajout de fichier à partir de l'interface web du gitlab : on peut tout faire à partir de la copie de travail et on est sûr alors de ne jamais rien faire d'irréparable.

Récupération de l'URL. La première opération s'effectue sur le gitlab avec un navigateur web. Ouvrir votre projet, cliquer sur `Clone` et choisir `Clone with HTTPS`. Copier l'URL du projet.

Création d'une copie de travail. L'opération suivante crée une copie de travail dans le répertoire courant. Il faut l'effectuer sur chaque ordinateur à partir duquel on souhaite travailler.

```
git clone <URL de mon projet>
```

3 Synchronisations futures avec le *projet initial*

Pendant le cours, il va falloir synchroniser votre projet avec le *projet initial*, qui va s'enrichir. Pour préparer ces synchronisations, on définit un *alias* pour le *projet initial*. Cet alias est : `projet-initial`.

Assurez-vous que le répertoire courant se situe à l'intérieur de la première copie de travail. La commande suivante définit un alias local à la copie de travail. Exécutez la commande suivante telle qu'elle est (copier-coller) :

```
git remote add projet-initial https://gitlab.univ-lille.fr/francois.boulier/is4-ano-2022.git
```

La commande suivante liste tous les alias définis dans votre copie de travail.

```
git remote -v
```

La commande suivante vous permettra, dans le futur, de synchroniser votre projet avec le projet initial. Il est possible qu'un éditeur de texte ouvre une fenêtre dans le terminal. Il suffit (normalement) de quitter cette fenêtre sans rien enregistrer.

```
git pull projet-initial main
```

4 Problème possible d'authentification

Certains étudiants ont des problèmes d'authentification bien qu'ils arrivent à se connecter au gitlab via l'interface web. Cela tient au fait qu'il y a *deux* mots de passe différents : un pour l'interface web et un pour l'authentification lorsque certaines commandes sont exécutées. Si vous rencontrez ce type de difficulté, connectez-vous au gitlab, accédez à votre compte en cliquant sur un bouton en haut à droite et changez le mot de passe de votre compte. Une solution simple consiste à mettre comme nouveau mot de passe du compte, celui qui est associé à votre identifiant univ-lille (terminé par "`.etu`").

5 Configuration de git

Pour éviter d'entrer trop souvent son identifiant et son mot de passe, il est possible de demander à `git` de les mémoriser pendant 15 minutes (la durée peut être configurée aussi)

```
git config --global credential.helper cache
```

Certaines commandes lancent un éditeur de texte mais il est souhaitable de choisir un éditeur de texte qui n'ouvre pas de fenêtre graphique : `nano` (à choisir si vous n'en connaissez aucun), `vim`, ou `emacs`.

```
git config --global core.editor <un éditeur>
```

La commande suivante rend certains affichages plus lisibles.

```
git config --global merge.conflictStyle diff3
```