université PARIS-SACLAY | FACULTÉ DES SCIENCES D'ORSAY

avril 2023

Outils de développement informatique
LDD2 MNSI

# Git

## Objectif

Le but de cette feuille est de :

1. se familiariser avec `git` et ses concepts, en ligne de commande
2. organiser le dépot du projet
3. commencer à prendre en main gitlab
4. s'organiser en groupe de projet et commencer à réfléchir

## 1 Prise en main de gitlab

Dans cette section, on se connecte à l'URL Gitlab de l'Université avec un navigateur Web.

1. Se connecter à `https://gitlab.dsi.universite-paris-saclay.fr` (prenom.nom/mot de passe du compte Paris-Saclay)
2. Noter son nom d'utilisateur gitlab (menu en haut à droite, `@prenom.nom` mais peut être différent)
3. **Un seul membre du groupe de projet** :
   — Menu → Group → Create Group
   — Créer un groupe
   — Aller sur le groupe, cliquer Invite Members
   — Ajouter les noms d'utilisateurs avec le rôle Maintainer
   — Ajouter `@kim.nguyen` avec le rôle Developer
   — Valider
4. **Le propriétaire du groupe** :
   — Crée un nouveau projet vide, privé, sans README.
5. Pour tout le monde, constater que le groupe/projet Git est visible dans l'interface Web

## 2 Génération d'une clé SSH

L'authentification avec le dépôt distant se fait de préférence en SSH.

1. Exécuter dans un terminal la commande

   ```
   $ cat ~/.ssh/id_rsa.pub
   ```

   Si le fichier n'existe pas faire l'étape suivante, sinon aller directement à l'étape 4.
2. Exécuter la commande `ssh-keygen`, accepter tous les choix avec Enter.
3. Refaire la commande 1
4. Copier la ligne `ssh-rsa ...` affichée dans le terminal (surligner/bouton droit copier)
5. Repartir sur l'interface Web `gitlab`
6. Dépiler votre menu de profil (en haut à droite) → Preferences
7. Dans la barre de gauche choisir, SSH Keys
8. Coller la ligne dans la zone de texte et sauver

## 3 Introduction à `Git`

Dans cette section, on effectue toutes les opérations `git` en ligne de commande dans un terminal.

1. Cloner le dépôt `git` de votre groupe :

   ```
   git clone git@gitlab.dsi.universite-paris-saclay.fr:votre-groupe/votre-projet.git
   ```

   Vous devez **modifier** `votre-groupe` par le nom de votre groupe et `votre-projet` par le nom du projet. Cette URL peut être récupérée directement depuis l'interface Web (menu Clone).

2. Un membre du groupe (désigné par le prof) crée un fichier texte `README.md` (ou le modifie s'il existe). Il y indique :

   ```
   # Projet XXXX

   ## Membres
     Foo Bar

   ## Description
     Todo
   ```

   où `XXXX` est le nom donné à votre projet et `Foo Bar` sont les noms de la personne. Faire ensuite `git add README.md` puis `git commit` avec un message raisonnable. Il faut ensuite faire `git push` pour envoyer le fichier sur le serveur. Constater que dans l'interface Web, le contenu du fichier `README.md` s'affiche joliment (le format `md` est du Markdown, dans lequel `#, ##, ###` sont des titres de section, `*foo*` met du texte en `gras`, ...).

3. Les autres membres font un `git pull` et éditent le fichier pour ajouter leur nom. Chacun fait `git commit -a README.md` (qui ajoute et `commit` tous les fichiers modifiés) puis `git push`. Que se passe-t'il ?

4. Régler les conflits. Chaque personne qui a eu un conflit, fait `git pull`, édite le fichier `README.md` pour retirer le conflit puis `git commit -a` suivi de `git push`.

## 4 Réflexions initiales

Commencez à discutter du projet en groupe. Vous pouvez prendre des notes (sur papier ou dans des fichiers). Les aspects technologiques (Java et BD) seront montrés en cours la semaine prochaine. Pour l'instant essayer de réfléchir sur une maquette de site, la façon dont l'utilisateur intéragit avec, les données à mémoriser et leur type. On essayera d'anticiper sur les fonctionalités avancées tant que possible (i.e. les ajouter aux fonctionalités initiales ne doit pas demander de tout reprendre à 0).

Pour la séance 2, tous les documents produits doivent se trouver dans un sous répertoire `doc/conception` du dépôt.