

TP 2 : Prise en main Git

Algorithmique – Programmation FIP (ING39)

V. Aponte, P. Courtieu

Septembre 2020

Ce Tp vous prépare à la méthode de travail à adopter TOUT LE TEMPS par la suite, ET AUSSI au premier Tp noté. Faites le consciencieusement.

MANTRA : *PROJET* GIT \neq *PROJET ECLIPSE*(/netbeans/vscode)

Relisez le paragraphe suivant jusqu'à l'avoir compris.

RÈGLE 1 : Jamais de **projet git** à l'intérieur d'un autre **projet git**.

RÈGLE 2 : Jamais de **projet Eclipse** à l'intérieur d'un autre **projet Eclipse**

RÈGLE 3 : Jamais **plusieurs projets git** dans un **projet Eclipse**.

RÈGLE 4 : **plusieurs projets Eclipse** dans un **seul projet git** OK mais sous-répertoires **complètement distincts**.

Sur le gitlab du CNAM (gitinfo.cnam.fr), vous pouvez créer des projets personnels. Comme vu en cours, il est possible, soit de créer un projet global où vous mettrez tous vos travaux sous Eclipse (sauf les TPs notés), soit de créer un projet Git par projet Eclipse. Nous faisons le choix suivant dans ce cours : nous avons créé pour vous un projet GIT où vous mettrez tous vos TP d'ING39 non notés. Il s'agit donc de respecter la règle 4 ci-dessus : un projet git contenant plusieurs projets Eclipse dans des sous-répertoires distincts. En particulier il ne faut pas créer un projet Eclipse à la racine du projet git.

1 Premier pas

Comme l'archive git ne contient aucun projet Eclipse pour l'instant, le plus simple est de cloner le répertoire depuis le terminal :

```
git clone https://gitinfo.cnam.fr/fip_2022/ing39_xxxx.git
```

Cette commande crée un répertoire du même nom que le projet Git.

1.0.1 Création du premier projet Eclipse

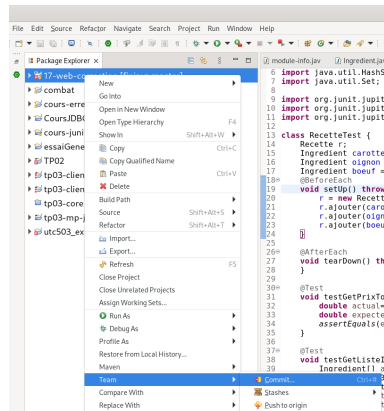
Vous avez récupéré votre archive, elle ne contient pas encore de projet Eclipse.

Cliquez sur File \Rightarrow new \Rightarrow Java project. Choisissez comme répertoire un SOUS-répertoire de l'archive `ing39-votrelogin/Tp0`. **Il faut donc créer ce sous-répertoire** avant de le sélectionner.

Votre projet Eclipse est créé, il faut y ajouter une classe principale : cliquez sur le répertoire `src` du projet, \Rightarrow new \Rightarrow class, appelez la `Main` et cochez la case `public static void main(...)`.

1.0.2 Un premier commit

Faites un premier commit de Tp0 : cliquez droit sur le projet Tp0 (important, ne pas cliquer sur un fichier) ⇒ Teams ⇒ Commit.



Sélectionnez les fichiers que vous voulez commiter (unstages ⇒ staged). La plupart du temps il faut sélectionner tous les fichiers créés ou modifiés.

Mettez un message de commit (obligatoire). ⇒ Commit. Tapez votre mot de passe éventuellement. Les petits sigles (>) qui indiquent que les dossiers n'ont pas été « commités » devraient disparaître.

Vos modifications sont maintenant conservées dans l'archive locale... **mais pas sur le serveur**. Pour les envoyer au serveur, il faut sélectionner Teams/Push to origin. Tapez vos identifiants, validez. Allez voir sur la forge si votre projet a bien été enregistré.

1.0.3 Le « vrai » premier tp sous Git (Tp2).

1. Créer un sous-répertoire tp2 dans le répertoire ing39-xxx
2. Choisissez le menu « File/New project ».
3. Sélectionner le sous-répertoire ing39-.../tp2 que vous avez créé.
4. Nommez le projet tp2 également. Lors de chaque Tp vous devez créer un nouveau projet Eclipse dans votre projet git et lui donner le nom TpN, où N est le Tp courant.
5. **commit. push.** Vérification que tout apparaît bien visible sur le site de la forge.

Exercice 1 : reprise des exercices du Tp1 sous Git via Eclipse

Dans cet exercice nous allons recopier le travail réalisé lors du Tp1 dans le projet Tp2 que vous venez de créer sous Git via Eclipse (voir plus haut). Cela vous permettra de terminer le Tp1 si ce n'était pas fait ! Pour trouver l'énoncé et squelette fournis pour le Tp1, voir la rubrique « Enoncés des Tps ».

Question 1

1. Ajoutez la classe ShiftTableau.java ainsi que le code qui implante la méthode decaleDroite en respectant son contrat.
2. Assurez vous que votre projet compile. Commit + push !
3. Ajoutez les jeux de tests nécessaires comme expliqué dans le Tp1.
4. Assurez vous, pour chaque ajout, que celui-ci compile. Plus précisément, à chaque fois qu'une nouvelle fonction ou test compile : COMMIT, puis PUSH !
5. Testez la méthode avec vos tests.

6. N'oubliez pas de commenter chacun de vos tests pour décrire le cas testé.
7. Et surtout **COMMIT ET PUSH!**

Question 2

Remplacez temporairement le code d'implantation de votre méthode par le code de votre voisin. Essayez de voir si vous pouvez y trouver des erreurs en jouant dessus vos tests!

Question 3

Si ce n'est pas déjà fait, attaquez l'exercice optionnel du Tp1 en suivant la même méthodologie.

A Gitlab

Marche à suivre pour créer un projet git sur gitlab :

- ouvrez un navigateur web ;
- connectez-vous sur la forge ;
- cliquez sur **New Project** ⇒ **create blank project** pour ajouter un nouveau projet ;
- Cochez la case **Initialize repository with a README**.
- Si vous voulez rendre le projet visible aux enseignants, mettez le dans le group **ing39_<votreannée>** (à ne pas confondre avec **ing39_<votreannée>_user**). Ou bien suivez les étapes ci-dessous.

Pour donner accès à votre projet aux enseignants :

- Sélectionnez votre projet, cliquez sur **Project information** puis **Members**
- ajoutez les utilisateurs Maria-Virginia Aponte, Serge Rosmorduc et Pierre Courtieu comme **Reporter** de votre projet (pour que nous puissions le lire).

B Eclipse

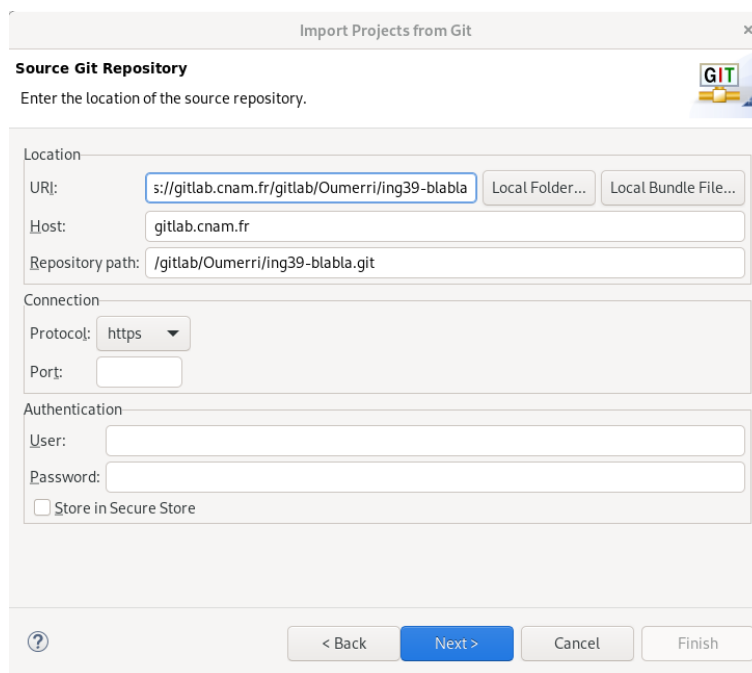
Cloner un projet depuis Eclipse

Ceci ne marche que si l'archive contient déjà au moins un projet Eclipse.

Menu **Files** ⇒ **Import** ⇒ **Project from git** ⇒ **clone URI**

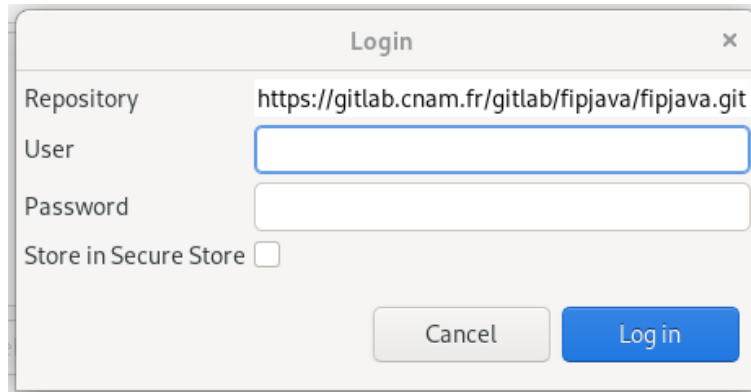
Remplissez le formulaire :

- URL du projet gitlab (**https** uniquement). L'URL de votre projet vous est donnée sur gitlab (projet nommé **ing39_votrelogin**). Les cases **Host** et **path** se remplissent automatiquement.
- **PAS** votre login tant que vous travaillez sur un compte local générique
- **PAS** votre mot de passe tant que vous travaillez sur un compte local générique



⇒ **Next**.

Tapez vos identifiant gitlab (mais ne cochez pas la case « store in Secure Store »).

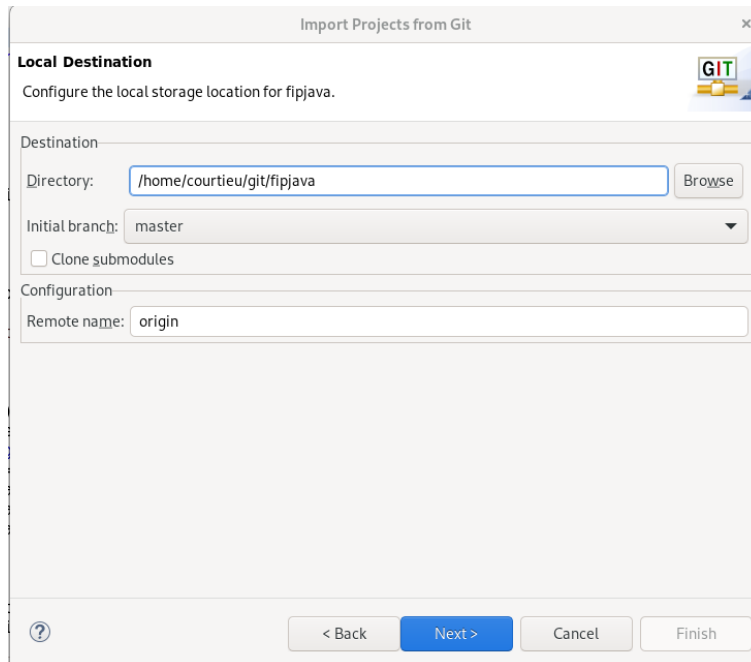


⇒ **Log in.**

2e formulaire : a priori ne touchez à rien.

⇒ **Next.**

3e formulaire : vous pouvez choisir l'emplacement local où cloner (télécharger) l'archive. NE PAS choisir un (sous-)répertoire d'une autre archive git (cf règle 1) ou d'un autre un projet Eclipse (règle 2).

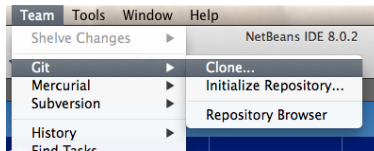


⇒ **Finish.**

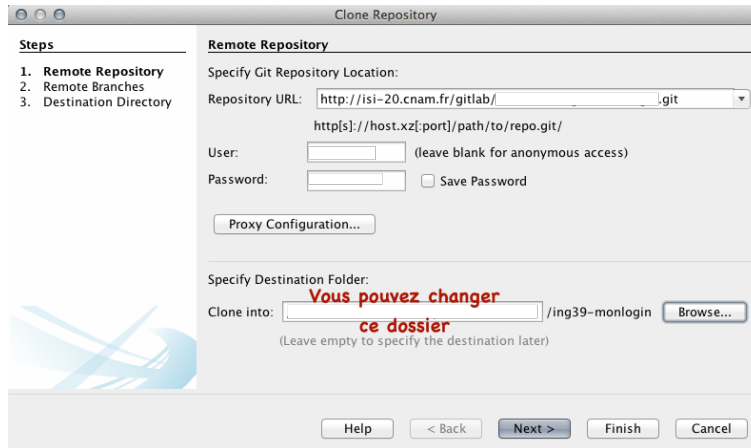
C Netbeans

Cloner un projet depuis Netbeans

Lancez Netbeans. Placez vous dans l'onglet **Services** de la fenêtre en haut à gauche. Cliquez sur **Team/Git/clone**

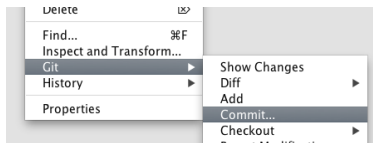


Remplissez le formulaire :



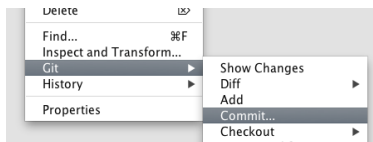
L'URL de votre projet vous est donnée sur gitlab. Cliquez sur finish, et acceptez la création d'un projet, que vous nommerez Tp0.

Pour faire un commit :

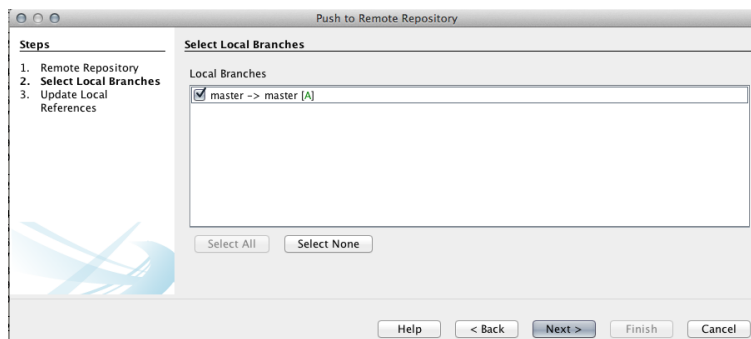


C.0.1 Commiter dans netbeanse

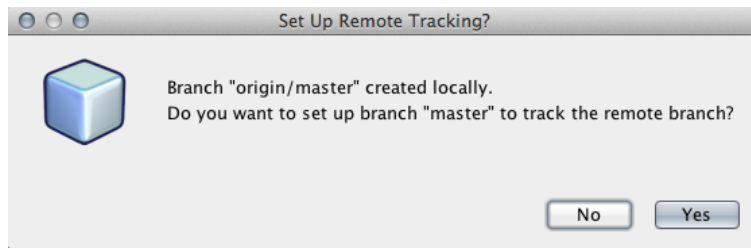
Faites un premier commit de Tp0,



Pour envoyer au serveur le commit, il faut sélectionner Git/Remote/Push. Une fenêtre nommée « Push to Remote Repository » apparaît. Cliquez sur **Next**.



Si la branche « master » n'est pas déjà cochée, cochez-là. Puis **Next** et **Finish**.
La fenêtre suivante s'ouvrira :



cliquez sur Yes. (attention, ne pas presser « entrée »)
Allez voir sur la forge si votre projet a bien été enregistré.