

Formation à Git

Exercice 1 : Création d'un compte et installation de Git

Création d'un compte git

Gitlab est un service cloud utilisant git. Gitlab dispose de fonctionnalités supplémentaires pour la collaboration, telles que le partage du code, la gestion de tâches ou le suivi des bugs. Dans ce TD, nous utiliserons la ForgeMIA (<http://forgemia.inra.fr>) qui est le nouveau système de gestion de développement collaboratif de logiciel du département de recherche Mathématiques et Informatique Appliquées (MIA) de l'Institut National de Recherche Agronomique (INRA). Cette forge, sous GitLab, est opérationnelle depuis début 2018, il suffit de disposer d'un compte INRA pour y avoir accès.

Installation de git sur votre machine

Vous pouvez vérifier si git est déjà installé en tapant la ligne de commande `git` dans votre terminal. Si la commande n'est pas reconnue entrez la commande suivante :

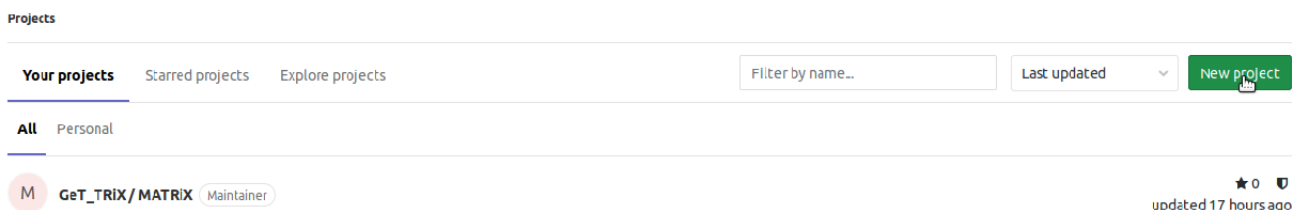
```
sudo apt-get install git
```

La commande suivante permet d'avoir un aperçu de toutes les fonctionnalités présentes dans le module git. Elles sont nombreuses mais uniquement certaines d'entre elles seront évoquées dans ce TD.

```
git help -a
```

Exercice 2 : Création d'un nouveau répertoire avec GitLab dans un groupe

- Allez sur GitLab et connectez-vous
- Cliquer sur votre icône en haut à droite → Profile → Groups
- Sélectionner le groupe dans lequel vous voulez ajouter un répertoire
- Cliquer sur New Project



<https://forgemia.inra.fr/>

Il faut dorénavant donner un nom au projet, définir sa visibilité, cliquer sur initialize with a README et enfin sur Create project.

Exercice 3 : Initialiser localement le répertoire git

Deux choix sont possibles avec la commande *remote* ou *clone* de git.

Remote

Pour la méthode *remote* afin de créer un référentiel local existant. Cette méthode permet de créer simplement une entrée dans votre configuration git qui spécifie un nom pour une URL particulière. Dans un premier temps il faut initier la racine du répertoire.

```
git init
```

```
git remote add origin https://gitlab.com/votreidentifiant/(nomdugroupe)/nomdurepo.git
```

Clonage

Pour la méthode *clone* est utilisé pour copier un dépôt existant situé à l'URL spécifiez

```
git clone https://forgemia.inra.fr/(nomdugroupe)/nomdurepo
```

Vous pouvez maintenant vous déplacer au sein de votre répertoire correspondant à la racine de votre projet.

```
cd nomdurepo
```

Exercice 4 : Premier commit

Un commit dans le langage *git* peut être assimilé à une *update* dans la branche correspondante *master* ou *définir*. On retrouve 3 étapes clés.

```
git add .
```

```
git commit -m "my first commit"
```

```
git push -u origin master
```

Ajouter un fichier *test.txt* avec les lignes de commandes ci-dessus dans la branche *master* de repository et visiter votre *repository* dans *forgemia* pour vérifier l'état du commit.

Exercice 5 : Ignorer les fichiers obsolètes

Il est possible de ne pas commit des fichiers en utilisant le fichier *.gitignore*. A partir de l'exercice précédent, créer un fichier *.gitignore* au niveau de la racine du dépôt. Ce dernier devra ne plus contenir les fichiers temporaires () et les fichiers R (*Rhistory* et *RData*) et compléter les informations manquantes.

fsoubes / MA_Trix_App

Watch

1

Star

1

Fork

0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

MA_Trix_App / .gitignore

or cancel

Want to use a .gitignore template?

Choose .gitignore: None

<> Edit file

Preview changes

Spaces

2

No wrap

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

History files

?

pas de fichier temporaire

?

Session Data files

?

Example code in package build process

*-Ex.R

Output files from R CMD build

/*.tar.gz

Output files from R CMD check

/*.Rcheck/

Coursera-Developing-Data-Products

RStudio files

.Rproj.user/

Compléter les ?