Notes de cours Git/GitHub

Thibaut Marmey

November 30, 2018

Contents

1 Git

1.1 Les commandes de bases

- Activer un repository Git, se placer dans le dossier que l'on veut et faire :

```
git init
```

- L'activation du repository Git génère un index. Lorsqu'on rajoute un nouveau fichier dans le repository il faut rajouter ce fichier à l'index par la commande :

```
git add nomDuFichier.extension
```

- Ajouter tous les fichiers du dossier courant :

```
git add .
```

- Revenir sur un *git add* , faire :

```
git reset
```

- Revenir en arrière après *git add*

```
git add --remove
```

- Pusher le commit

```
git commit -m "ajout de fichier"
//le -m permet de rajouter un message au commit
```

- Pour se positionner sur un commit de l'historique :

```
git checkout SHADuCommit
```

- Revenir au commit le plus récent :

```
git checkout master
```

- Créer un nouveau commit qui fait l'inverse du commit précédent :

```
git revert SHADuCommit
```

- Modifier le message du dernier commit

```
git commit --amend -m "votre message"
```

- Annuler tous les changements qui ne sont pas encore commités :

```
git reset --hard
```

- Pour créer une nouvelle branche, se placer au commit avec un checkout et faire :

```
git checkout -b nouvelle-branche
```

- Pour supprimer une branche, se placer à l'extérieur de celle-ci et fait :

```
git branch -d nom-de-la-branche
```

- Copier un repository depuis GitHub

```
git clone adresseDuRepository
```

- Utilisation de *Rebase* :

```
git rebase master bug
```

  - Transplante bug sur l'actuelle branche master. Si on est déjà en train de bosser sur bug1 on peut se contenter de taper

    ```
    git rebase master
    ```

  - Switch sur master

    ```
    git checkout master
    ```

    S
  - fusionne bug1 dans master

```
git merge bug1
```

  - fusionne bug1 dans master

```
git branch -d bug1
```

- Savoir qui a modifé un fichier :

```
git blame nomDuFichier.extension
```

- Voir les détails du commit :

```
git show numSHA
```

- Ignorer des fichier dans Git telles que des variables de configuration, des mots de passe, etc :
  utiliser un fichier *.gitignore*. Dans ce fichier, indiquer leur chemin complet et sauter une ligne entre chaque fichier.

- Eviter les conflits superflus.

  - En cas "d'urgence", si la tâche actuelle n'est pas finie et donc ne nécessite pas de commit, mais qu'une autre tâche doit être faite, il faut mettre de côté ce qui est entrain d'être fait, pour pouvoir y revenir après. Utiliser :

```
git stash //permet de mettre en pause les modifications pour
    travailler sur un autre endroit du projet
```

  - Pour revenir à l'endroit où vous vous étiez arrétés, ici

```
pop
```

  - Va effacer le stash donc si l'on veut refaire une pause, il faut refaire

```
git stash pop
```

  - Pour garder les modifications dans le stash, utiliser :

```
git stash apply
```

2 GitHub

2.1 Les commandes de bases

- GitHub est aussi appelé un remote, il permet de sauvegarder ses fichiers sur un espace distant.

- Configurer compte avec GitHub :

```
git config --global user.name "tmarmey"
git config --global user.email "tmarmey@mail.com
```

- Envoyer ses modifications sur GitHub faire :

```
git push origin master
```

- Ajouter clef SSH pour push sans identifiant et mot de passe :
  - *$ ssh-keygen -t rsa -b 4096 -C "yourEmail@example.com"*
  - Enter a file in which to save the key (/home/you/.ssh/id_rsa): [Press enter]
  - Start the ssh-agent : $ eval "$(ssh-agent -s)"
  - $ ssh-add /.ssh/id_rsa
  - Ajouter la clef au compte GitHub
  - copier la clef avec xclip : $ sudo apt-get install xclip
  - $ xclip -sel clip ¡ ~/.ssh/id _rsa.pub
  - Settings/SSH and GPG keys
  - New SSH key et Add SSH key
  - $ ssh -T git@github.com
- Récupérer les dernier modifications :

```
git pull origin master
```

- Contribuer à un projet :
  - Fork le repo de l'auteur (c'est à dire copier le repo sur votre compte GitHub)
  - cloner le fork sur votre machine
  - Créer une nouvelle branche
  - Faire des modifs sur cette nouvelle branche
  - Envoyer les modifs sur GitHub sur votre propre branche
  - Proposer votre contribution au projet par "pull request"