

Git : mode d'emploi pour un usage seul, avec dépôt distant, sur une seule branche

Florian Legendre

Université de Poitiers

Année 2020 - 2021

Table of Contents

- 1 Synchroniser son travail
- 2 Configurer le suivi d'une branche distante
- 3 La boucle de travail multibranches connectées

Table of Contents

1 Synchroniser son travail

- Récupérer les modifications d'un serveur distant
- Ajouter ses mises à jour sur un serveur distant

Commandes : git fetch / git pull

git fetch	git pull
<ul style="list-style-type: none">❶ Récupère seulement la dernière mise à jour de l'historique❷ Il faut ensuite faire un git merge pour incorporer les changements❸ Utile quand on veut anticiper des conflits ou qu'on n'a pas tout à fait fini son travail	<ul style="list-style-type: none">❶ Récupère seulement la dernière mise à jour de l'historique et incorpore les changements simultanément❷ Elle est plus utilisée que git fetch

Commande : git push

La commande "git push" permet de mettre à jour la branche distante en incorporant dans la branche distante l'historique et les fichiers locaux.

ATTENTION :

- ❶ Il faut avoir fait un git pull avant de l'utiliser
- ❷ Il faut avoir les droits (cf. Module sur la collaboration avec Git)

Table of Contents

- 2 Configurer le suivi d'une branche distante
 - Cloner un projet
 - Récupérer les autres branches distantes du projet cloné
 - Créer une branche distante depuis Git
 - Suivre une branche distante

Commande : git clone <url>

```
crex@crex:~/projects$ git clone https://github.com/torvalds/linux.  
git  
Cloning into 'linux'...  
remote: Enumerating objects: 7886755, done.  
remote: Total 7886755 (delta 0), reused 0 (delta 0), pack-reused  
7886755  
Receiving objects: 100% (7886755/7886755), 2.98 GiB | 1.12 MiB/s,  
done.  
Resolving deltas: 100% (6548657/6548657), done.  
Updating files: 100% (71277/71277), done.
```

Listing 1 – Exemple de git clone

Récupérer les autres branches distantes du projet cloné

Quand vous récupérez un projet cloné vous récupérez aussi ses autres branches. Seulement elles sont cachées :

```
crex@crex:~/projects/tmp/Banquise-Game(master)$ git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/dev
remotes/origin/master
```

Listing 2 – Afficher les branches cachées

Récupérer les autres branches distantes du projet cloné

Pour travailler sur ces branches et configurer automatiquement leurs suivis vous pouvez vous rendre directement dessus avec un *git checkout* *<nomBranche>*. Attention ! Le *<nomBranche>* doit être exactement celui de la branche distante :

```
crex@crex:~/projects/tmp/Banquise-Game(master)$ git checkout dev
Branch 'dev' set up to track remote branch 'dev' from 'origin'.
Switched to a new branch 'dev'
```

Listing 3 – Suivre une branche distante du projet

Créer une branche distante depuis Git

Mon conseil : ne passez pas par GitHub pour créer une branche distante. Dans Git, vous créez une branche (ou si elle existe déjà vous vous y rendez) et vous entrez la commande *git push* *<aliasURL> <nomBranche>* :

```
crex@crex:~/projects/tmp/Banquise-Game(dev)$ git checkout -b exp
Switched to a new branch 'exp'
crex@crex:~/projects/tmp/Banquise-Game(exp)$ git branch -a
dev
* exp
master
remotes/origin/HEAD -> origin/master
remotes/origin/dev
remotes/origin/master
crex@crex:~/projects/tmp/Banquise-Game(exp)$ git push origin exp
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote:
To github.com:Chuxclub/Banquise-Game.git
* [new branch]      exp -> exp
```

Listing 4 – Ajouter une branche distante au projet depuis Git



Commandes : `git remote add <al> <url>` / `git remote rm <al>`

La première commande "git remote add" suivi d'un alias et d'une url permettent de suivre la branche distante désignée par l'url.

Un alias est un comme un pseudo donné à l'url il permet :

- 1 De spécifier où faire le push d'une branche :
`git push <al> <nomBrancheDistante>`
- 2 De spécifier quelle branche du dépôt distant on veut pull :
`git pull <al> <nomBrancheDistante>`
- 3 De supprimer un suivi de branche distante :
`git remote rm <al>`
- 4 De spécifier une branche où on veut push/pull par défaut :
`git push --set-upstream <al> <nomBrancheDistante>`

Commande : git remote -v

Comme pour les branches locales on peut vouloir savoir quelle branche distante on suit actuellement ou encore l'ensemble des alias vers des url que contient notre branche locale. Ces informations sont accessibles par la commande *git remote -v*.

Exemple :

```
crex@crex:~/projects/GitLearn(main)$ git remote -v
origin  git@github.com:Chuxclub/GitLearn.git (fetch)
origin  git@github.com:Chuxclub/GitLearn.git (push)
```

Listing 5 – Exemple d'output de la commande git remote -v

Table of Contents

3 La boucle de travail multibranches connectées

La boucle de travail multibranches connectées

