

Configurer git

Lancer la machine

Configuration de git

Installation

Préparation de windows

- Télécharger git pour windows depuis <https://git-scm.com/downloads>.
- Installer git pour windows dans le chemin par défaut.
- Choisir "Use git from the Windows Command Prompt"
- Choisir "Checkout as is, commit as-is".

Préparation de RStudio

- Menu Tools > Global Option > git/SVNs.
- Vérifier que git est activé et que le chemin vers l'exécutable git est le bon. (sous windows : `C:/Program Files/Git/bin/git.exe`)

Configuration de git

Depuis le terminal de Rstudio (*alt shift R*), sont définis les paramètres généraux de configuration de git. Ces instructions ne sont à exécuter qu'une fois sur la machine.

Qui nous sommes :

```
git config --global user.name "vincent Guyader"  
git config --global user.email votre@email.fr
```

Le proxy :

git n'utilisera pas la configuration proxy de votre Rstudio, il faut le lui donner explicitement.

```
git config --global http.proxy pfrie-std.proxy.e2.rie.gouv.fr:8080  
git config --global https.proxy pfrie-std.proxy.e2.rie.gouv.fr:8080
```

Configuration de git

Winmerge est un outil graphique utile, entre autres pour la résolution des conflits de merge

```
git config --global merge.tool winmerge
git config --global mergetool.prompt false
git config --global mergetool.winmerge.trustExitCode true
git config --global mergetool.winmerge.cmd "\"C:\\Program Files (x86)
```

Pour éviter de taper sans cesse son mot de passe, on peut aussi demander à git de le mémoriser :

```
git config --global credential.helper store
```

Configuration de git

L'instruction `git config --global` retourne le contenu de la configuration (*faire défiler au besoin avec les flèches et taper sur la lettre q pour sortir*)

```
$ git config --global --list
user.email=vincent@thinkr.fr
user.name=Vincent Guyader
merge.tool=winmerge
mergetool.winmerge.cmd="C:\\Program Files (x86)\\WinMerge\\WinMergeU.exe" -u -e -dl Local -dr Remote "$LOCAL" "$REMOTE" "$MERGED"
mergetool.winmerge.trustexitcode=true
mergetool.prompt=false
credential.helper=store
```

il s'agit du contenu du fichier `.gitconfig` qui se trouve dans le home de l'utilisateur. Au besoin, il peut-être édité avec un éditeur de fichier texte.

Configuration git

Initialisation d'un projet R avec git

Le principe de fonctionnement repose sur le fait qu'à chaque projet R est associé un projet git. Il faut posséder un compte sur le **remote** qui va héberger vos projets versionnés.

Fabriquer un projet sur gitlab

- Se connecter sur <http://git.thinkr.fr/> (avec les identifiants fournis)
- Utiliser le bouton **New Project** pour créer un projet.
- Choisissez un nom pour votre projet :
 - en minuscule
 - sans point, espace ou underscore
 - qui ne commence pas par un chiffre

Dans la suite de ce document nous feront référence a ce projet sous le nom **"monprojet"**. Par commodité, ce nom doit correspondre au nom de votre projet R existant ou à venir.

Configuration git

Initialisation d'un projet R avec git

Vous devez choisir, quand vous fabriquez un projet, sous quel Namespace vous le placez. Par défaut il sera construit dans votre Namespace utilisateur : `http://git.thinkr.fr/votre_nom/monprojet` . Vous pouvez le placer dans un groupe `http://git.thinkr.fr/un_groupe/monprojet` , rendant, en fonction de la configuration de gitlab, votre projet accessible aux autres membres du groupe.

Configuration git

Initialisation d'un projet R avec git

Récupérez alors le lien `http` vers votre projet en cliquant sur le bouton `clone` :
`http://git.thinkr.fr/votre_nom/monprojet.git`

votre_nom > monprojet > Details

Project 'monprojet' was successfully created.






monprojet 
Project ID: 333

 [Add license](#)  **0** Commits  **0** Branches  **0** Tags  **0** Bytes Files


The repository for this project is empty

If you already have files you can push them using the [command line instructions](#) below.

Note that the master branch is automatically protected. [Learn more about protected branches](#)

  Star **0** [Clone](#) 

Clone with SSH

`ssh://git@git.thinkr.fr:2222,` 

Clone with HTTP

`http://git.thinkr.fr/votre_nom` 

Configuration git

Initialisation d'un projet R avec git - Cas d'un Nouveau Projet Rstudio

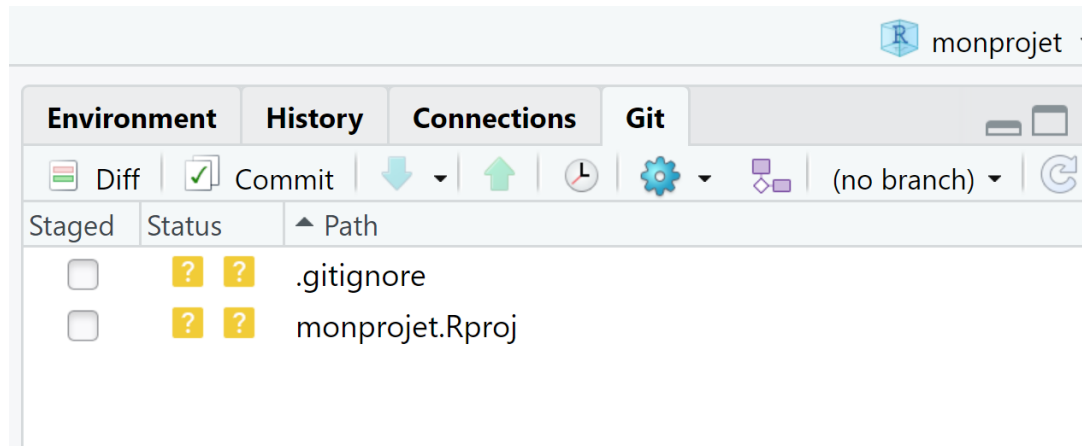
Dans Rstudio :

- Menu File > New Project
- Choisir : Version Control > git
- Repository URL : `http://git.thinkr.fr/votre_nom/monprojet.git`
- Project Directory Name : doit correspondre à *monprojet*
- Create project as subdirectory : Choisir l'emplacement du projet sur votre ordinateur (
~ correspond a votre 'home' c'est-à-dire "Mes Documents" sous Windows)

Configuration git

Initialisation d'un projet R avec git - Cas d'un Nouveau Projet Rstudio

Dans ce nouveau projet Rstudio créé vous verrez apparaître l'onglet git en haut à droite



Configuration git

Initialisation d'un projet R avec git - Cas d'un ancien projet Rstudio

- Ouvrez le projet Rstudio et lancez dans la console l'instruction `usethis::use_git()`. (le package *usethis* doit etre installé)
- Répondez par l'affirmative à toutes les questions posées dans la console. Un premier commit "Initial Commit" est fait pour vous.
- Rstudio redémarre et git est opérationel en local.

Il vous faut maintenant relier ce projet à votre **remote**, dans le terminal, saisir :

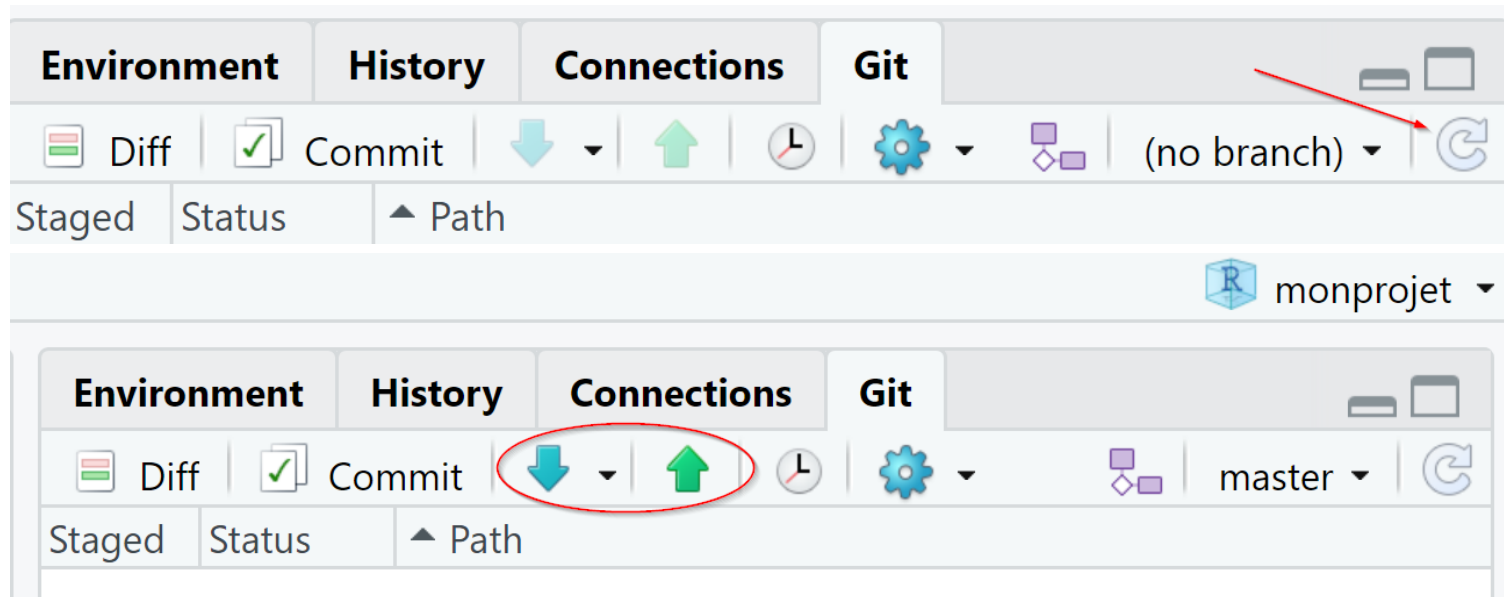
```
git remote add origin 'http://git.thinkr.fr/votre_nom/monprojet.git'  
git push -u origin master
```

vous venez de rajouter un **remote** et de faire un premier **push**.

Configuration git

Initialisation d'un projet R avec git - Cas d'un ancien projet Rstudio

Si les boutons **push** et **pull** (les 2 flèches verticales) du panneau git restent grisés, cliquez sur la flèche circulaire de rafraîchissement du panneau.



Configuration git

Vérifier la configuration

Depuis un terminal, vérifiez la localisation du **remote** :

```
git remote --verbose
```

On peut ajouter un projet distant avec :

```
git remote add remote-name URL
```

ou le supprimer avec

```
git remote rm remote-name
```