

GIT avec Github

Table des matières

Git	2
Installation de Git.....	2
Branches	5
Github.....	7
Travail collaboratif.....	8
Avec GIT GUI	9
Création d'une clé SSH	10
Cloner avec SSH	10



Local
Version Control System



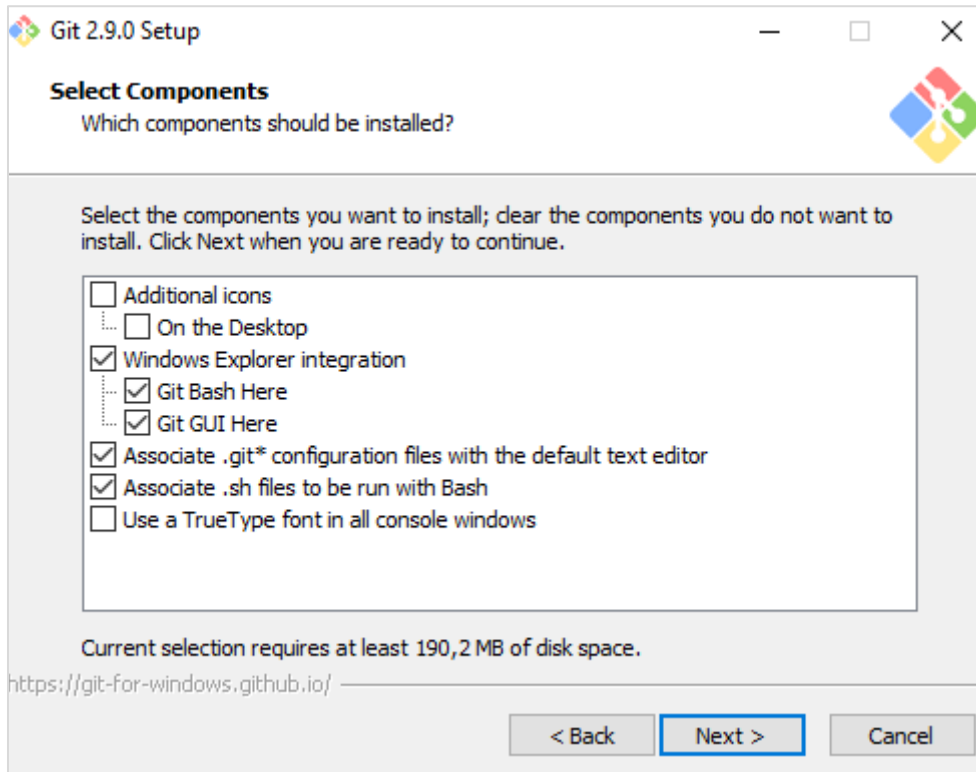
Web

Git

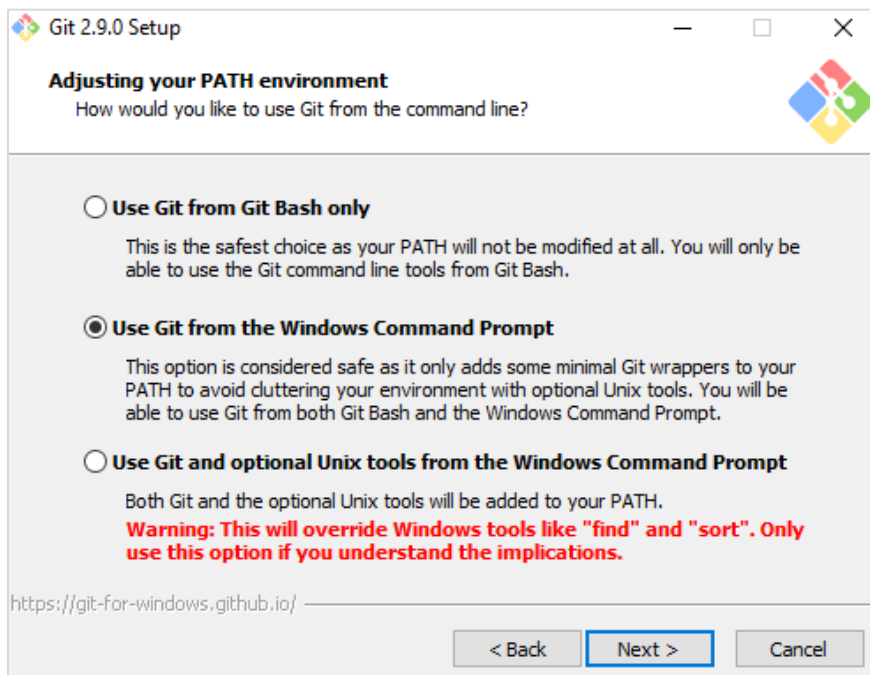
[Documentation](#)

Installation de Git

<https://www.git-scm.com/download/win>



Utilisation de git bash et invite de commande Windows ?



Configuration du nom et email de l'utilisateur

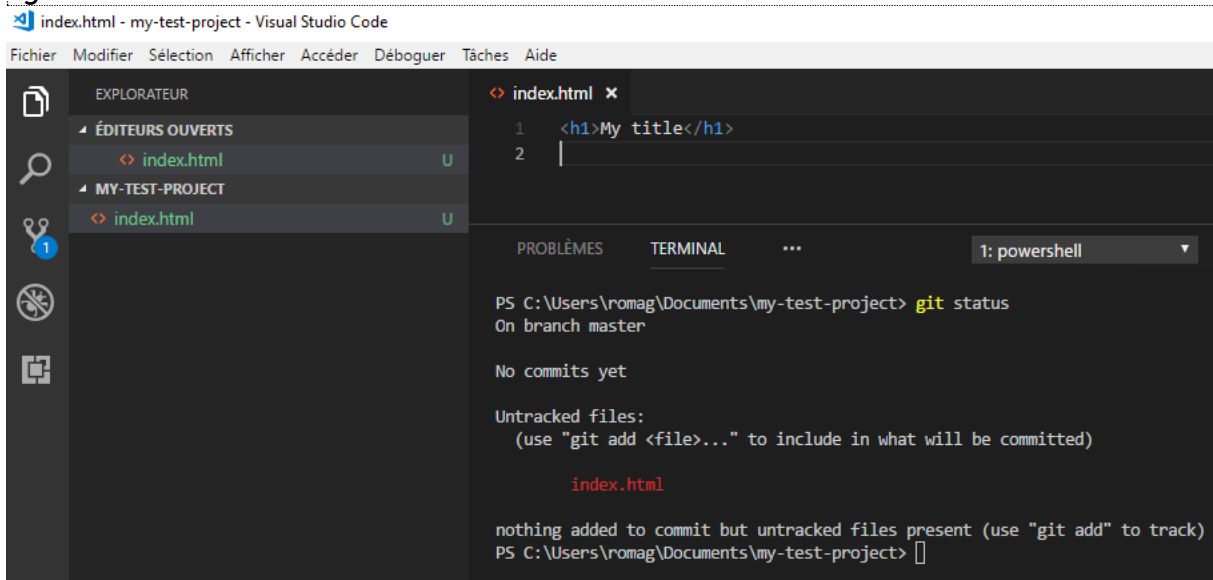
```
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"
```

Initialisation

```
git init
```

Voir le statut des fichiers

```
git status
```



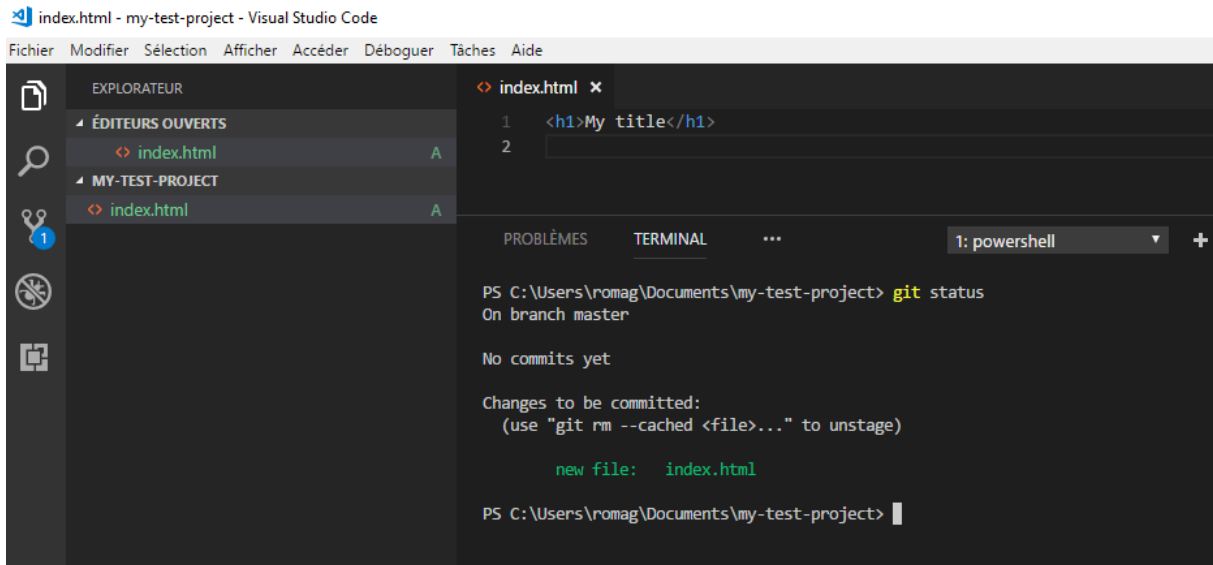
The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left shows a project named 'MY-TEST-PROJECT' with an 'index.html' file. The Editor window shows the content of 'index.html' as '<h1>My title</h1>'. The Terminal window at the bottom shows the output of the 'git status' command:

```
PS C:\Users\romag\Documents\my-test-project> git status  
On branch master  
  
No commits yet  
  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
  
      index.html  
  
nothing added to commit but untracked files present (use "git add" to track)  
PS C:\Users\romag\Documents\my-test-project>
```

Valider les changements

```
git add .
```

(statut après)



Commit

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

```
PS C:\Users\romag\Documents\my-test-project> git commit -m "first commit"
[master (root-commit) 3138156] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 index.html
PS C:\Users\romag\Documents\my-test-project> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\romag\Documents\my-test-project> |
```

Il suffit ensuite de répéter cette boucle pour chaque ajout, suppression, modification de fichiers :

```
git add .
git commit -m "my description"
```

Afficher l'historique des commits avec **"git log"**

```
PS C:\Users\romag\Documents\my-test-project> git log
commit 568a48084af4b431d35c11f83964b2e0bc3125aa (HEAD -> master)
Author: Jerome Romagny <romagny13@yahoo.fr>
Date: Sun Jun 24 23:29:56 2018 +0200

    style.css and paragraph added

commit 31381560419ac8fe764cb6f90c4afcbec65dc367
Author: Jerome Romagny <romagny13@yahoo.fr>
Date: Sun Jun 24 23:25:52 2018 +0200

    first commit
PS C:\Users\romag\Documents\my-test-project> |
```

On peut ajouter des options, exemple avec stat

```
PS C:\Users\romag\Documents\my-test-project> git log --stat
commit 568a48084af4b431d35c11f83964b2e0bc3125aa (HEAD, master)
Author: Jerome Romagny <romagny13@yahoo.fr>
Date: Sun Jun 24 23:29:56 2018 +0200

    style.css and paragraph added

index.html | 1 +
style.css  | 3 +++
2 files changed, 4 insertions(+)
```

« Rollback »

Exemple on retourne au premier commit

```
git checkout 31381560419ac8fe764cb6f90c4afcbec65dc367
```

```
PS C:\Users\romag\Documents\my-test-project> git log
commit 568a48084af4b431d35c11f83964b2e0bc3125aa (HEAD, master)
Author: Jerome Romagny <romagny13@yahoo.fr>
Date: Sun Jun 24 23:29:56 2018 +0200

    style.css and paragraph added

commit 31381560419ac8fe764cb6f90c4afcbec65dc367
Author: Jerome Romagny <romagny13@yahoo.fr>
Date: Sun Jun 24 23:25:52 2018 +0200

    first commit
PS C:\Users\romag\Documents\my-test-project> git checkout 31381560419ac8fe764cb6f90c4afcbec65dc367
Previous HEAD position was 568a480 style.css and paragraph added
HEAD is now at 3138156 first commit
PS C:\Users\romag\Documents\my-test-project> |
```

Ou avec reset

```
git reset --hard 31381560419ac8fe764cb6f90c4afcbec65dc367
```

(options : hard pour suppression, mixed, soft)

Branches

Création d'une branche nommée « my-branch »

```
git checkout -b my-branch
```

Voir les branches avec

```
git branch
PS C:\Users\romag\Documents\my-test-project> git branch
master
* my-branch
```

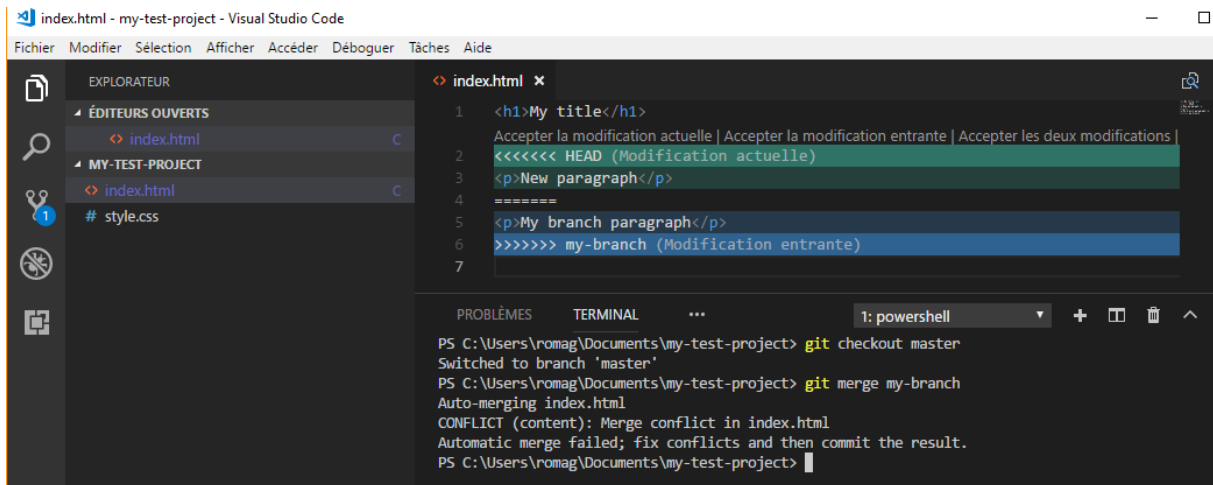
Switch de branche, exemple switch vers la brancne "master"

```
git checkout master
```

```
PS C:\Users\romag\Documents\my-test-project> git checkout master  
Switched to branch 'master'
```

Merge

```
git merge my-branch
```



```
index.html - my-test-project - Visual Studio Code  
Fichier Modifier Sélection Afficher Accéder Débuguer Tâches Aide  
EXPLORATEUR  
ÉDITEURS OUVERTS  
  index.html C  
MY-TEST-PROJECT  
  index.html C  
  # style.css  
index.html x  
1 <h1>My title</h1>  
Accepter la modification actuelle | Accepter la modification entrante | Accepter les deux modifications |  
2 <<<<<< HEAD (Modification actuelle)  
3 <p>New paragraph</p>  
4 =====  
5 <p>My branch paragraph</p>  
6 >>>>>> my-branch (Modification entrante)  
7  
PROBLÈMES TERMINAL  
1: powershell  
PS C:\Users\romag\Documents\my-test-project> git checkout master  
Switched to branch 'master'  
PS C:\Users\romag\Documents\my-test-project> git merge my-branch  
Auto-merging index.html  
CONFLICT (content): Merge conflict in index.html  
Automatic merge failed; fix conflicts and then commit the result.  
PS C:\Users\romag\Documents\my-test-project> |
```

Supprimer une branche

```
Git branch -D my-branch
```

```
PS C:\Users\romag\Documents\my-test-project> Git branch -D my-branch  
Deleted branch my-branch (was e2b918d).  
PS C:\Users\romag\Documents\my-test-project> git branch  
* master
```

Search or jump to... Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: romagny13 / Repository name: my-test-project ✓

Great repository names are short and memorable. Need inspiration? How about [special-happiness](#).

Description (optional): My test project description

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None Add a license: None ⓘ

Create repository

romagny13 / my-test-project Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH `https://github.com/romagny13/my-test-project.git`

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# my-test-project" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/romagny13/my-test-project.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/romagny13/my-test-project.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

ProTip! Use the URL for this page when adding GitHub as a remote.

Se connecter à un projet

```
git remote add origin https://github.com/romagnyl3/my-test-project.git
```

Push des modifications (demande nom et mot de passe utilisateur Github)

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

The screenshot shows the GitHub interface for a repository named 'my-test-project' by user 'romagnyl3'. At the top, there are statistics: 1 watch, 0 stars, and 0 forks. Below this, there are navigation tabs for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, and Settings. The main content area shows a description 'My test project description' with an 'Edit' button. Below the description, there are statistics: 4 commits, 1 branch, 0 releases, and 1 contributor. There are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A commit history table is visible, showing three commits: 'First commit' (4 minutes ago), 'Branch paragraph added' (22 minutes ago), and 'style.css and paragraph added' (an hour ago). Below the commit history, there is a preview of the README.md file, which contains the text 'My test project'.

Travail collaboratif

Clone

```
git clone https://github.com/romagnyl3/my-test-project.git
```

Naviguer vers/ ouvrir le dossier copié ... faire des modifications

Voir les branches remote

```
PS C:\Users\romag\Documents\my-clone-project\my-test-project> git remote  
origin
```

```
PS C:\Users\romag\Documents\my-clone-project\my-test-project> git branch -r  
origin/HEAD -> origin/master  
origin/master
```

Peut être utile pour connaître le nom afin de merge une branche

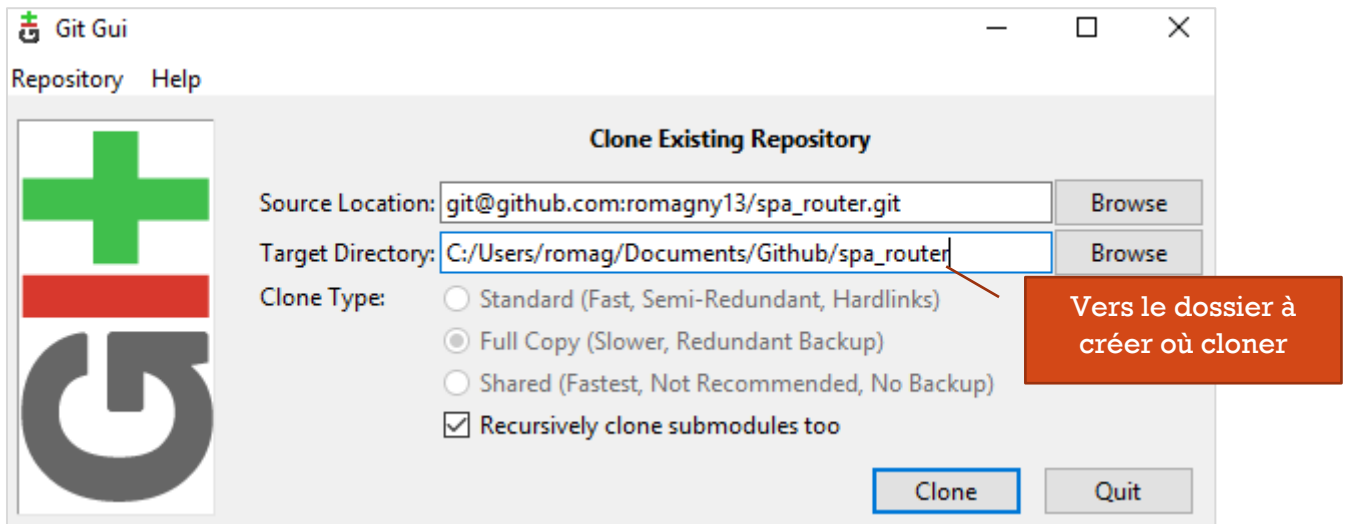
Puis **push**

```
git push
```


Remote rollback, exemple

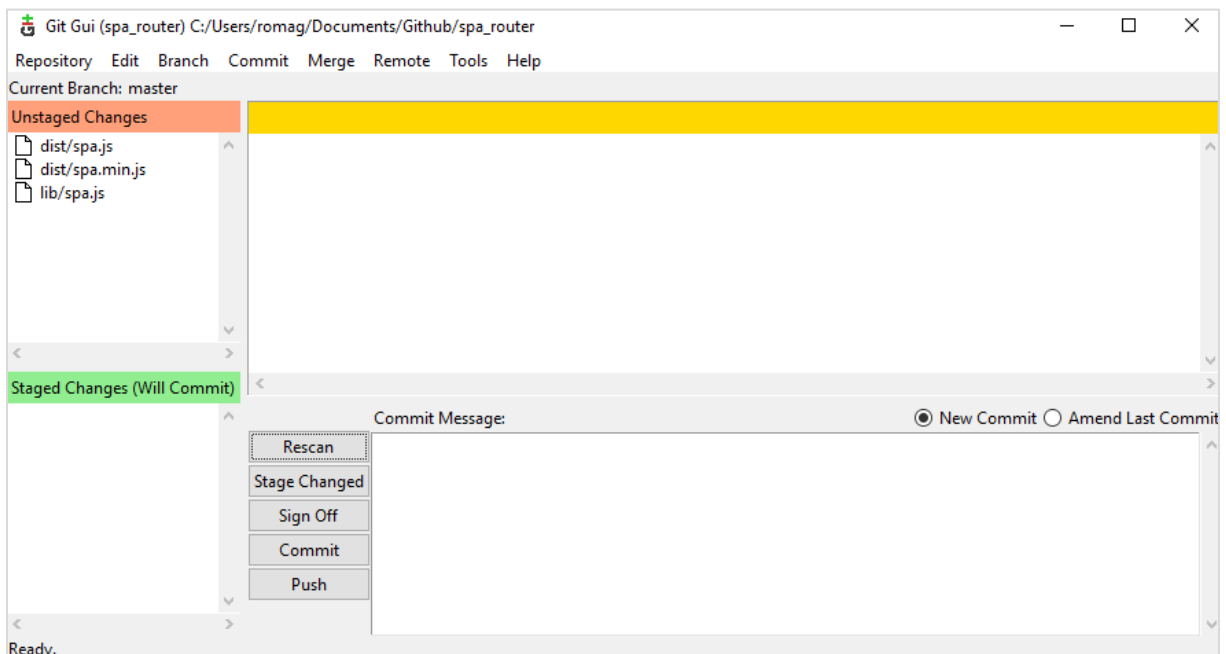
```
git reset --hard 0ef2c8c559c0e03f8fdedb332129c193e998169c  
git push -f origin
```

Avec GIT GUI



Après avoir fait des changements (ajout fichier, etc.)

- **Rescan** pour voir les changements
- **Stage changed ...**
- **Commit** en indiquant un **message** dans la boîte de texte à droite des boutons
- Et enfin **PUSH** pour uploader les fichiers modifiés.



Création d'une clé SSH

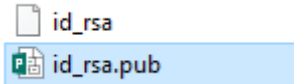
Aide

Ouvrir Gitbash

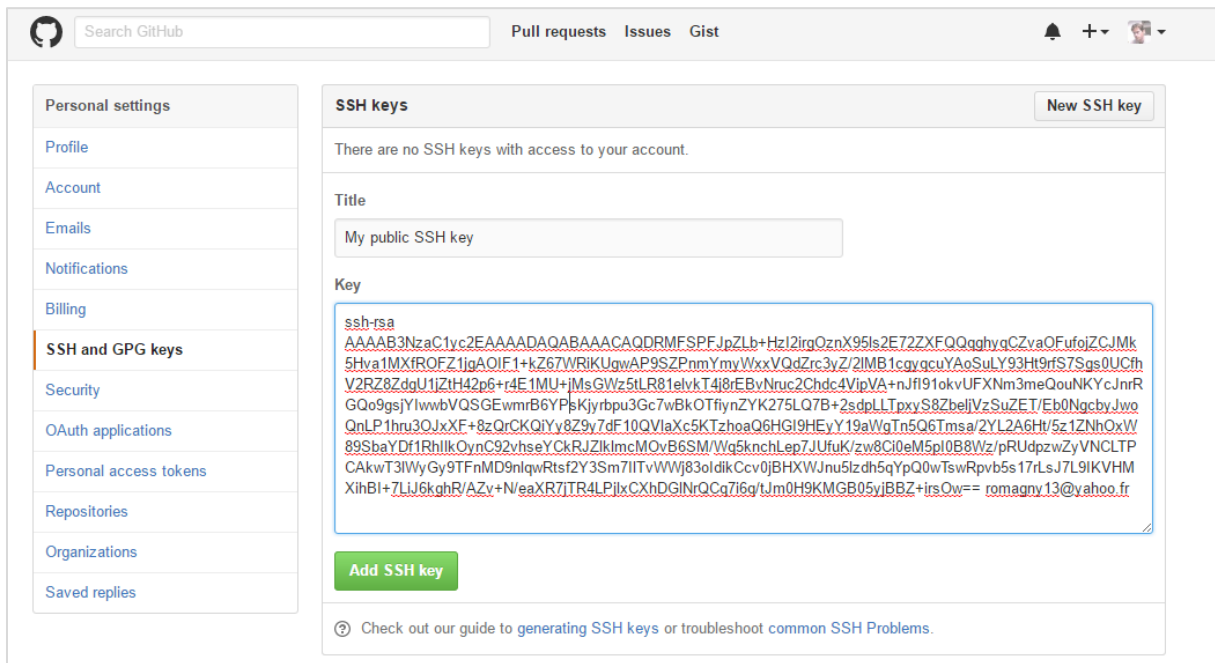
```
ssh-keygen -t rsa -b 4096 -C "romagny13@yahoo.fr"
```

La clé est ajouté au répertoire désiré (créé s'il n'existe pas) (exemple « C:\Users\[user]\.ssh »)

Récupérer la clé publique dans le répertoire



Dans les paramètres du compte sur **Github** ... éditer le profil ... onglet SSH



Cloner avec SSH

Récupérer / créer un nouveau répertoire sur Github

