

# TRAVAILLER EN ÉQUIPE



# ► **Travailler en équipe**

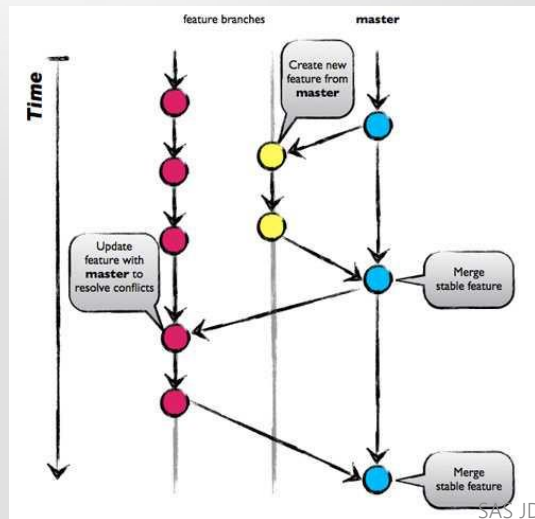
Workflow



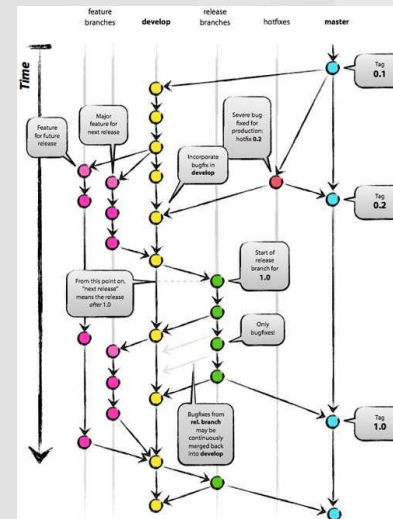
## Workflow : une organisation

- ≥ Des règles à suivre pour travailler ensemble
- ≥ Plusieurs méthodes existes selon votre projet → Vous pouvez créer votre propre workflow  
<https://www.nicoespeon.com/fr/2013/08/quel-git-workflow-pour-mon-projet/>

### 1 - GitHub-Flow

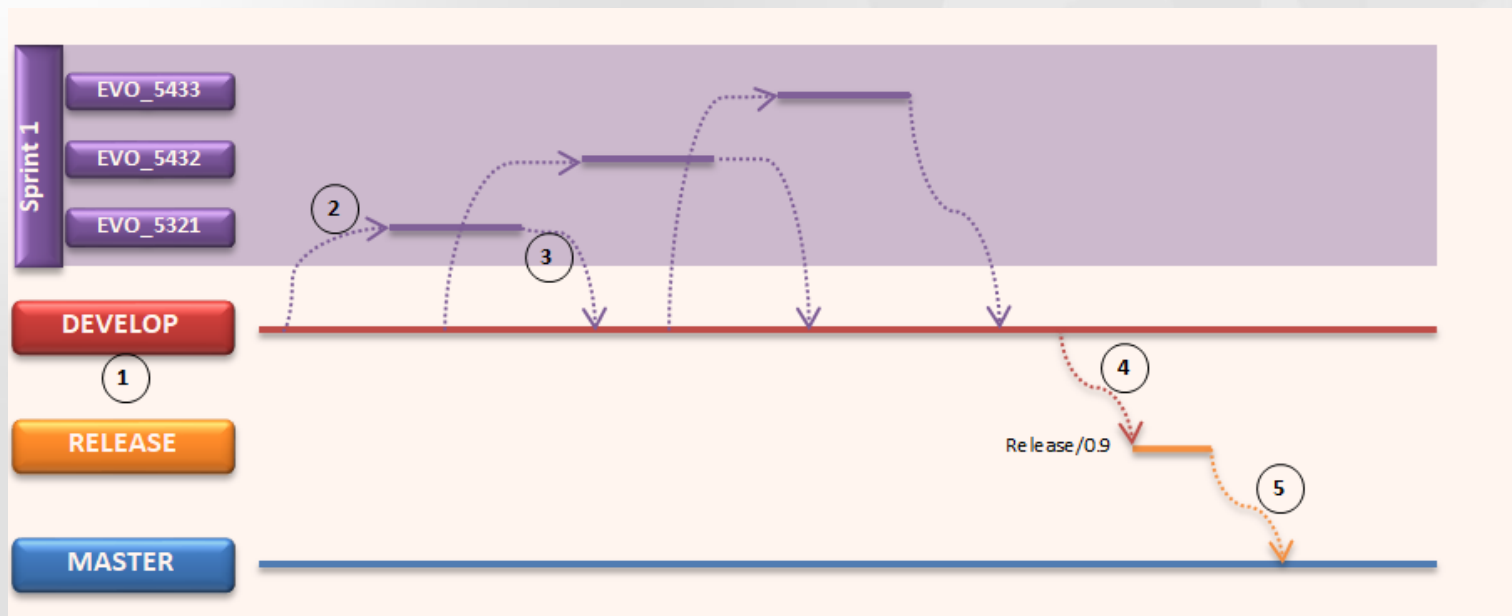


### 2 - Git-Flow



## Workflow : une organisation

- ≥ Revue du workflow Rennes Métropole



# ► **Travailler en équipe**

Maintenir son fork



### Préparation

1. Définir une branche **immuable** (par défaut master)
2. Définir un dépôt distant de référence (via l'URL utilisée pour le git clone du projet)

```
MINGW64:/c/Users/gaetan/Documents/git/mviewer
gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (santegraphie)
$ git remote -v
origin  https://github.com/jdev-org/mviewer.git (fetch)
origin  https://github.com/jdev-org/mviewer.git (push)

gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (santegraphie)
$ git remote add upstream https://github.com/geobretagne/mviewer

gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (santegraphie)
$ git remote -v
origin  https://github.com/jdev-org/mviewer.git (fetch)
origin  https://github.com/jdev-org/mviewer.git (push)
upstream https://github.com/geobretagne/mviewer (fetch)
upstream https://github.com/geobretagne/mviewer (push)

gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (santegraphie)
$ |
```

#### Conseil

Basez-vous sur le workflow du projet mère de votre fork !



### Mettre à jour la branche immuable

#### 3. Mettre à jour la branche immuable (master)

```
gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (master)
$ git fetch upstream
remote: Enumerating objects: 523, done.
remote: Counting objects: 100% (523/523), done.
remote: Compressing objects: 100% (47/47), done.
remote: Total 1188 (delta 489), reused 499 (delta 475), pack-reused 665
Receiving objects: 100% (1188/1188), 518.20 KiB | 1.46 MiB/s, done.
Resolving deltas: 100% (774/774), completed with 112 local objects.
From https://github.com/geobretagne/mviewer
```

```
* [new branch]      3d -> upstream/3d
* [new branch]      bug-428 -> upstream/bug-428
* [new branch]      develop -> upstream/develop
* [new branch]      elasticsearch7 -> upstream/elasticsearch7
* [new branch]      feature-109 -> upstream/feature-109
* [new branch]      feature-155 -> upstream/feature-155
* [new branch]      feature-176 -> upstream/feature-176
* [new branch]      feature-277 -> upstream/feature-277
* [new branch]      issue-364 -> upstream/issue-364
* [new branch]      issue-427 -> upstream/issue-427
* [new branch]      master -> upstream/master
* [new branch]      pr-issue-386 -> upstream/pr-issue-386
* [new branch]      satellite -> upstream/satellite
* [new tag]          v3.4.1 -> v3.4.1
* [new tag]          v3.3 -> v3.3
* [new tag]          v3.4 -> v3.4
```

```
gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (master)
$ git reset --hard upstream/master
Checking out files: 100% (3558/3558), done.
HEAD is now at 68611d5 Merge pull request #412 from geobretagne/develop
```

```
gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (master)
$ git push origin master --force
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/jdev-org/mviewer.git
32e51e1..68611d5 master -> master
```

On se place sur la branche master locale via la commande :  
`$ git checkout master`

Récupération des modifications depuis le dépôt parent

La branche master locale devient identique à l'état de la branche master du dépôt parente

On publie l'état local vers le serveur



## Update des branches de développement

4. Faite une nouvelle branche depuis votre branche à mettre à jour (sauvegarde via GitHub)
5. Réalisez un rebase (cf. la doc Mviewer commanditée par Rennes Métropole)

[https://mviewerdoc.readthedocs.io/fr/latest/doc\\_contrib/git.html#rebase](https://mviewerdoc.readthedocs.io/fr/latest/doc_contrib/git.html#rebase)

### Conseil

Utilisez un éditeur de texte pour  
gérer les conflits !





## Cas spécifique : git-flow

- ≥ Avec le workflow git-Flow
- ≥ Nécessite 2 branches immuables : master + develop
- ≥ Ici un exemple pour récupérer dans sigrennesmetropole/mviewer une branche develop-contrib immuable de la branche geobretagne/mviewer develop

```
gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/RM-mviewer/mviewer (master)
```

```
$ git branch -D develop-contrib
```

```
Deleted branch develop-contrib (was 0b013469).
```

Supprimer la branche si elle existe déjà

```
gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/RM-mviewer/mviewer (master)
```

```
$ git checkout -b develop-contrib upstream/develop
```

```
Switched to a new branch 'develop-contrib'
```

```
Branch 'develop-contrib' set up to track remote branch 'develop' from 'upstream'.
```

Création de la branche local qui suivra la branche develop de geobretagne/mviewer

```
gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/RM-mviewer/mviewer (develop-contrib)
```

```
$ git push -u origin develop-contrib
```

```
Total 0 (delta 0), reused 0 (delta 0)
```

```
remote:
```

```
remote: Create a pull request for 'develop-contrib' on GitHub by visiting:
```

```
remote: https://github.com/sigrennesmetropole/mviewer/pull/new/develop-contrib
```

```
remote:
```

```
To https://github.com/sigrennesmetropole/mviewer.git
```

```
* [new branch]
```

```
Branch 'develop-contrib' set up to track remote branch 'develop-contrib' from 'origin'.
```

C'est prêt !

## Head détachée

- ≥ Etat sur une branche inexistante (temporaire)
- ≥ Pour éviter d'être en Head détachée ➔ Créez une branche comme ici avec un tag :

`git checkout -b v2.9.X`

# ► **Travailler en équipe**

Pull Request



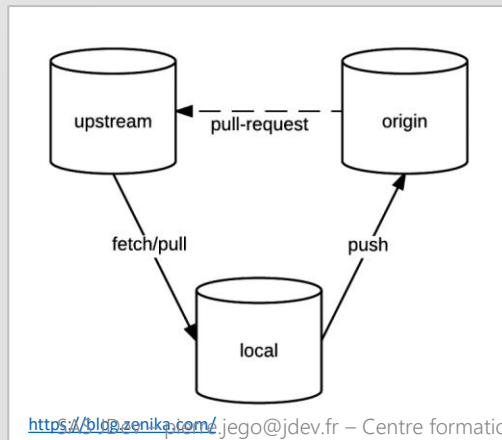
### C'est quoi ?

Définition :

<https://blogs.infinisquare.com/posts/alm/pullrequest-mais-pourquoi-on-s-embete-a-en-faire>

Une PR, requête de tirage ou Pull Request est une demande de relecture d'un code par d'autres développeurs afin d'intégrer du code d'une branche sur une autre branche.

La branche contenant le code à intégrer peut être sur le même dépôt ou sur un dépôt différent (fork).



### Risques

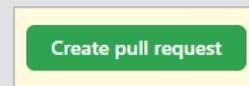
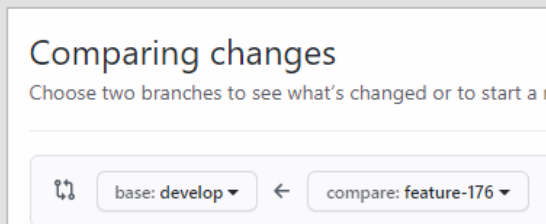
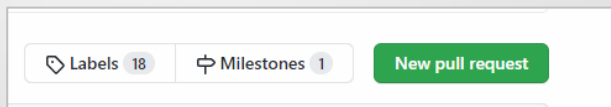
- ≥ Nouveaux bugs
- ≥ Régressions
- ≥ Arbre des commits complexe à lire
- ≥ Importer de la complexité dans le code
- ≥ Importer des failles de sécurité, des ressources obsolètes

#### Conseil

Ne pas hésitez à partir d'une branche clean pour faire la PR !



### Comment on fait ?



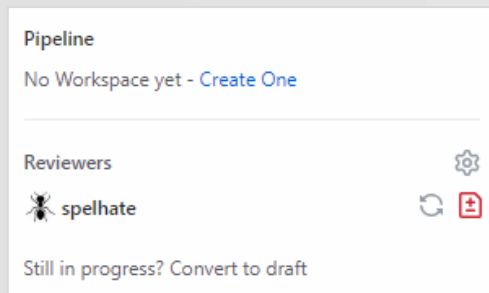
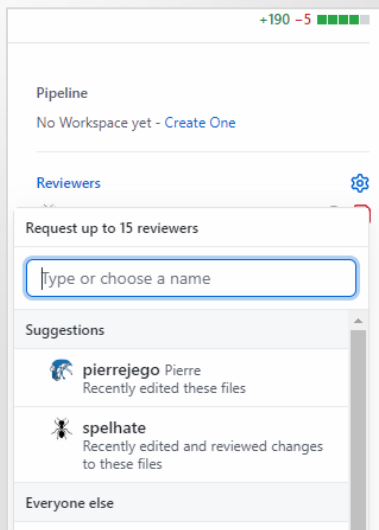
#### Conseil

Vérifiez bien vos branches !  
Refaites rapidement une passe sur  
le code avant de créer la PR...



### Affecter des relecteurs (reviewers)

- ≥ Permet désigner et notifier des relecteurs (option)



#### Conseil

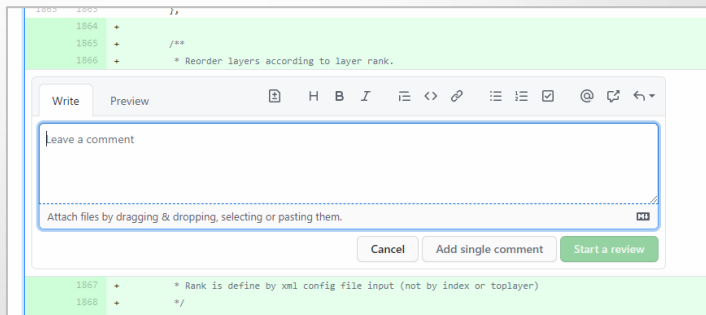
Sélectionner les relecteurs selon leur maîtrise technique et l'intérêt du projet





### Effectuer une revue de code

- ≥ En tant que relecteur vous pouvez ajouter des commentaires, demander des modifications ou valider la PR
- ≥ Une relecture implique de tester la pull request !



#### Conseil

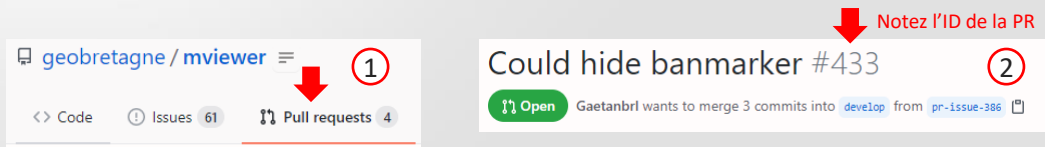
Soyez précis et explicite dans vos relectures !  
L'erreur est également permise...



### Tester une pull request

- ≥ Si vous devez gérer des PR, vous devrez alors les tester localement
- ≥ Pour cela, on crée une nouvelle branche locale qui contiendra le code de la branche source de la PR  
<https://docs.github.com/en/free-pro-team@latest/github/collaborating-with-issues-and-pull-requests/checking-out-pull-requests-locally>

**Prérequis :** Vous devez avoir défini le dépôt de la PR dans vos remotes !



```
gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (master)
$ git remote -v
origin https://github.com/jdev-org/mviewer.git (fetch)
origin https://github.com/jdev-org/mviewer.git (push)
upstream https://github.com/geobretagne/mviewer (fetch)
upstream https://github.com/geobretagne/mviewer (push)

gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (master)
$ git fetch upstream pull/433/head:pr-433
From https://github.com/geobretagne/mviewer
* [new ref]      refs/pull/433/head -> pr-433

gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (master)
$ git checkout pr-433
Switched to branch 'pr-433'
```

Utilisez l'ID de la PR



Passez sur la  
branche



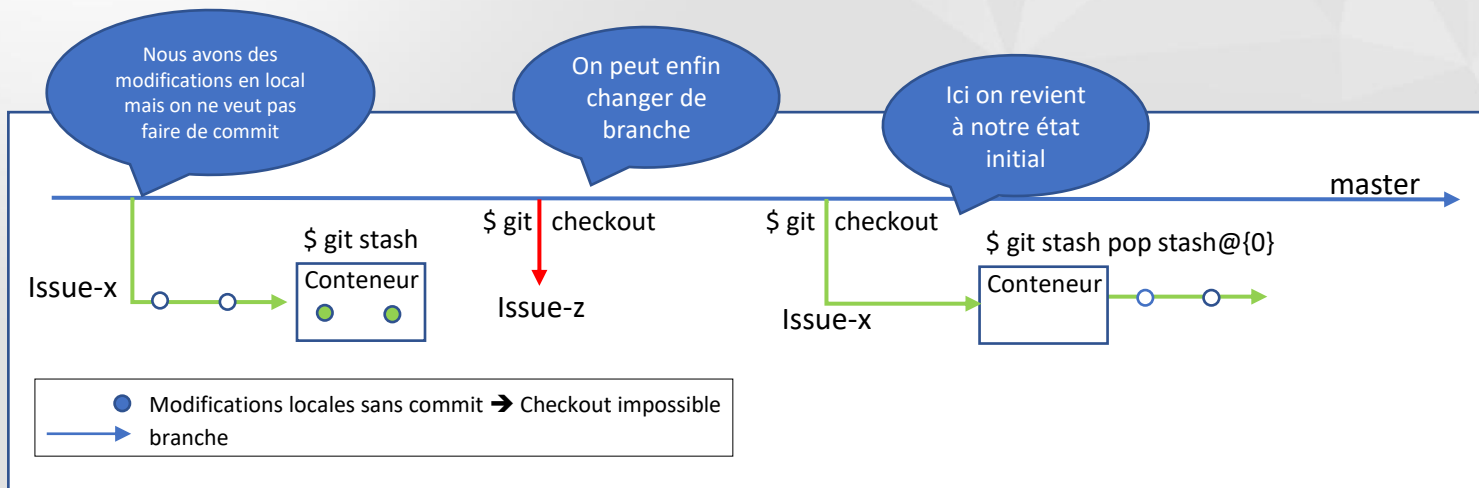
# ► **Travailler en équipe**

L'entreposage



### Git Stash : l'entreposage

- ≥ Vous devrez souvent jongler entre les branches (ex: hotfix sur production)
- ≥ L'entreposage permet d'enregistrer toutes les modifications en cours d'une branche dans un conteneur (en attente)



### Création + récupération

MINGW64: c:/Users/gaetan/Documents/git/mviewer

```
gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (pr-433)
$ git status
On branch pr-433
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (pr-433)
$ git stash
Saved working directory and index state WIP on pr-433: 7cf391b Fix dubble trigger for showLocation

gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (pr-433)
$ git status
On branch pr-433
nothing to commit, working tree clean

gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (pr-433)
$ git stash pop stash@{0}
On branch pr-433
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
Dropped stash@{0}: (bcf90ff43d047658ab01d9e918404b07cb9bad4b)

gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (pr-433)
$ git status
On branch pr-433
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (pr-433)
$ |
```

### Création + suppression

```
gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (pr-433)
$ git status
On branch pr-433
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")

gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (pr-433)
$ git stash
Saved working directory and index state WIP on pr-433: 7cf391b Fix dubble trigger for showLocation
g
gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (pr-433)
$ git stash list
stash@{0}: WIP on pr-433: 7cf391b Fix dubble trigger for showLocation

gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (pr-433)
$ git stash drop stash@{0}
Dropped stash@{0} (5c82890534f1af3507698e794cc4ee9ba4c078a4)

gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (pr-433)
$ git status
On branch pr-433
nothing to commit, working tree clean

gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (pr-433)
$ |
```

# ► **Travailler en équipe**

Memento



### Commandes utiles

≥ Squash, changement de message, suppression de commit, etc...

```
git rebase -i ID_COMMIT
```

```
git push -f
```

≥ Stash

```
git stash
```

```
git stash list
```

```
git stash pop stash@{0}
```

```
git stash drop stash@{0}
```

≥ Supprimer une branche locale

```
git branch -D NOM_BRANCHE
```

≥ Supprimer une branche locale

```
git branch -D NOM_BRANCHE
```

≥ Retourner sur un commit antérieur

```
git checkout ID_commit
```

```
git push -f
```



### Commandes utiles

≥ Ecraser l'état local par l'état distant (fichiers de l'index compris)

`git reset --hard HEAD`

≥ Mettre un fork à jour (branche master)

`git remote add upstream https://github.org/sigrennesmetropole/repo.git`

`git fetch upstream`

`git checkout master`

`git reset --hard upstream/master`

`git push origin master --force`

≥ Rebase → `git rebase [branche à ramener] [branche à mettre à jour]`

`git checkout master`

`git rebase origin/master issue-125`

≥ Récupérer les Sous module (clone déjà fait)

`git submodule update --init`

≥ Récupérer les Sous module pendant le clone

`git clone --recurse-submodules https://github.com/`

### Commandes utiles

- ≥ Créer un sous-module (création d'un fichier .gitmodules)

```
git submodule add https://github.com/foo/bar.git
```

```
git commit -m 'nouveau sous module'
```

```
git push origin/master
```

- ≥ Copier / coller un commit d'une branche vers une autre

```
git cherry-pick ID_COMMIT
```

- ≥ Tester une pull request

```
git fetch origin pull/ID/head:BRANCHNAME
```

- ≥ Vérifier l'arbre des commits

```
git log
```

...En mode graphique

```
git log --oneline --decorate --graph --all
```



```
graph TD
    68a152e["68a152e (issue-386) Fix case when showMarker param is not set - work with previous version"]
    c62c983["c62c983 Fix #386 and hide or display banmarker"]
    68611d5["68611d5 (tag: v3.4.1, origin/master, origin/HEAD, jdev/master, master) Merge pull request #412 from geobretagne/develop"]
    95e6779["95e6779 (tag: v3.4) Merge pull request #403 from geobretagne/develop"]
    32e51e1["32e51e1 (tag: v3.3) Merge pull request #336 from geobretagne/develop"]
    88b19a7["88b19a7 (refs/stash) WIP on develop: 17a70fc Fix overlay on mouse over - if no features"]
    4e28b02["4e28b02 index on develop: 17a70fc Fix overlay on mouse over - if no features"]
    b037d2d["b037d2d (origin/issue-427, jdev/issue-427) Hack bad contentType send by QGIS server GetFeatureInfo. See #427"]
    3054412["3054412 Remove trailing spaces"]

    68a152e --- c62c983
    c62c983 --- 68611d5
    68611d5 --- 95e6779
    95e6779 --- 32e51e1
    32e51e1 --- 88b19a7
    88b19a7 --- 4e28b02
    4e28b02 --- b037d2d
    b037d2d --- 3054412
```

### Commandes utiles

≥ Travailler avec des tags

<https://git-scm.com/book/fr/v2/Les-bases-de-Git-%C3%89tiquetage>

# ► **Travailler en équipe**

Les sous-module



### Explications

- ≥ Les sous-modules ou « submodules » permettent d'intégrer un dépôt Git au sein d'un dépôt Git
- ≥ Ils permettent ainsi de gérer de façon autonomes des dépôts enfants comme des projets indépendants (communauté indépendantes, versions indépendantes, issues indépendantes, etc...)
- ≥ Il faut utiliser l'option « --recurse-submodules » lors du clone d'un dépôt pour récupérer les sous-modules d'un projet

```
$ git clone --recurse-submodules https://github.com/chaconinc/MainProject
```

Besoin d'aller plus loin ?

<https://git-scm.com/book/fr/v2/Utilitaires-Git-Sous-modules>

# ► **Travailler en équipe**

Maîtriser les issues GitHub

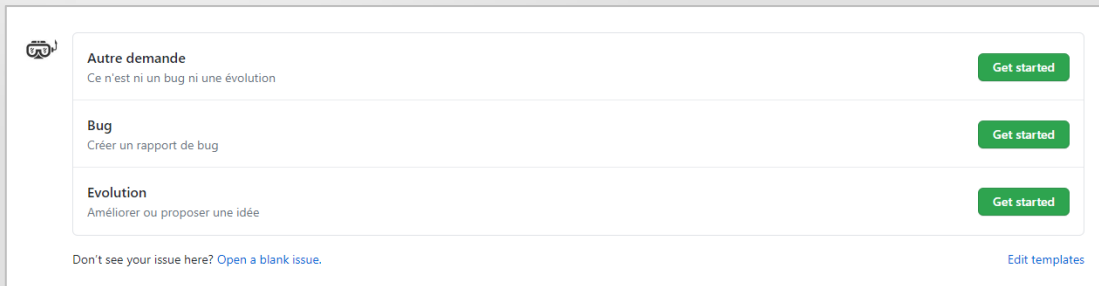


### Les issues

- ≥ Permettent de publier une question, un bug, un sujet de discussion, une demande d'évolution, etc...



- ≥ Des modèles (template) peuvent être faciliter la rédaction



- ≥ Les templates sont à ajouter dans à la racine dans `.github/ISSUE_TEMPLATE`  
[https://github.com/geobretagne/mviewer/tree/master/.github/ISSUE\\_TEMPLATE](https://github.com/geobretagne/mviewer/tree/master/.github/ISSUE_TEMPLATE)











### Les issues et le markdown

- ≥ Les issues utilisent la syntaxe Markdown et disposent d'un mode prévisualisation

**Issue: Bug**

Créer un rapport de bug. If this doesn't look right, [choose a different type](#).

 Bug sur la toolbar de dessin

Write Preview H B I       

**\*\*Description du bug / Describe the bug\*\***

Une description courte du problème


**\*\*Reproduire le bug / To Reproduce\*\***

Etapes à suivre pour reproduire votre problème du type :



**Issue: Bug**

Créer un rapport de bug. If this doesn't look right, [choose a different type](#).

 Bug sur la toolbar de dessin

Write Preview

Description du bug / Describe the bug

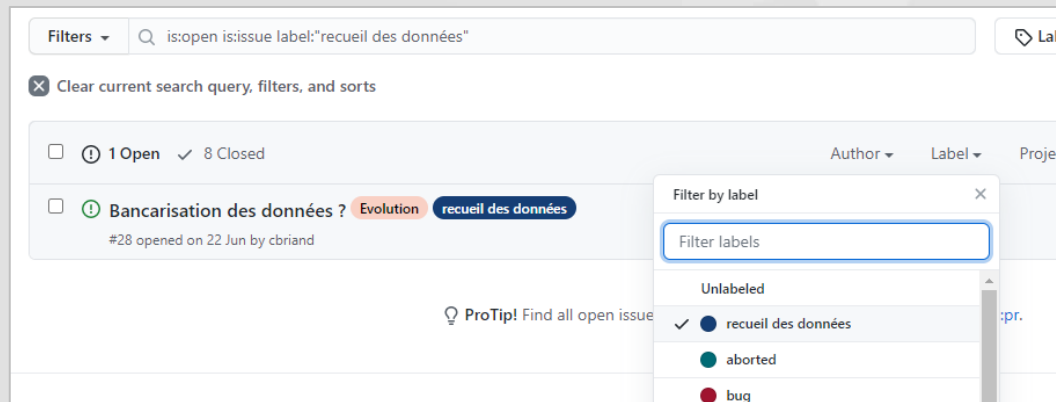
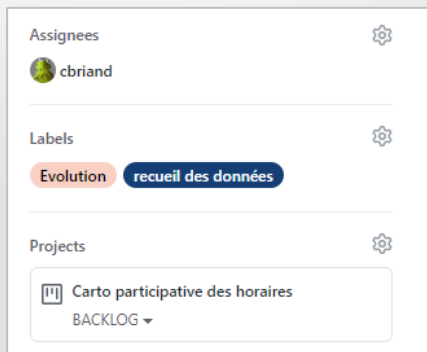
Une description courte du problème

Reproduire le bug / To Reproduce

Etapes à suivre pour reproduire votre problème du type :

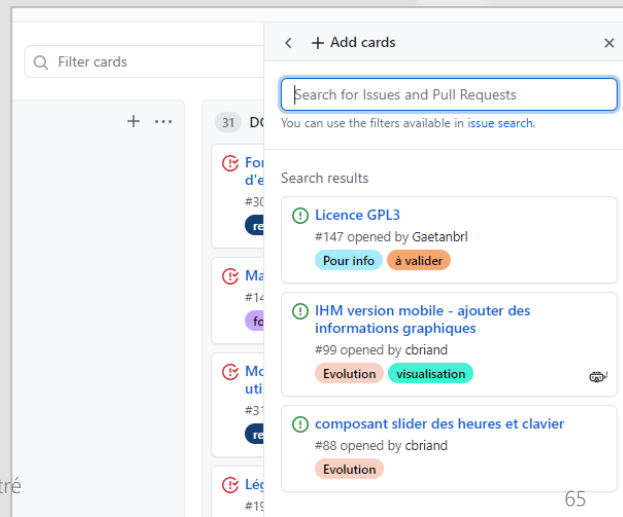
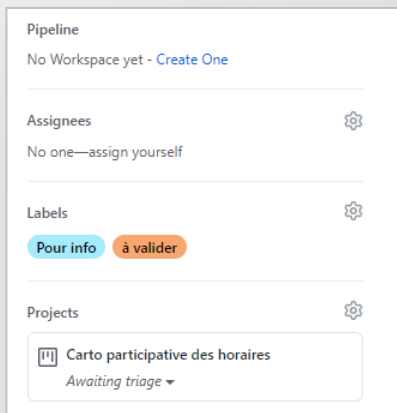
### Les issues

- ≥ Une issue peut disposer d'informations propres
- ≥ Elle peut être reliées à un compte, une version, un projet
- ≥ Les labels sont utiles pour catégoriser une issue et faciliter la recherche



### Projets et issues

- ≥ Les issues peuvent être utilisées comme des histoires utilisateur en mode Agile
- ≥ La partie Projet permet alors de les manipuler
- ≥ Prérequis : Un projet doit être affecté à une issue !



# ► **Travailler en équipe**

Authentification



## Gestion locale de l'authentification

≥ La commande « git config » permet de gérer l'authentification pour un dépôt (--local) ou pour tous les dépôts (--global)

1. On peut avoir une authentification par dépôt

```
$ cd git/mviewer  
$ git config --local user.name "gaetanbrl"  
$ git config --local user.email gaetan.brue1@jdev.fr
```

2. Ou avoir une authentification globale à tous les dépôts via l'option git « --global » et selon l'utilisateur de l'OS

```
$ git config --global user.name "gaetanbrl"  
$ git config --global user.email gaetan.brue1@jdev.fr
```

3. Il est possible d'affecter une authentification pour l'ensemble d'un système d'une machine complète

```
$ git config --system user.name "gaetanbrl"  
$ git config --system user.email gaetan.brue1@jdev.fr
```

## Gestion locale de l'authentification

- ≥ On peut lister les valeurs de la configuration via l' paramètre « --list »

```
gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (pr-433)
$ git config --list
core.symlinks=true
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
```

### Gestion locale de l'authentification

- ≥ Il est aussi possible de définir un fichier de configuration git qui est localisé pour un projet spécifique dans le répertoire .git/ sous le nom « config »

```
gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (pr-433)
$ git config --list
core.symlinks=true
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
gaetan@DESKTOP-1HU3HJF MINGW64 ~/Documents/git/mviewer (pr-433)
$ nano .git/config |
```

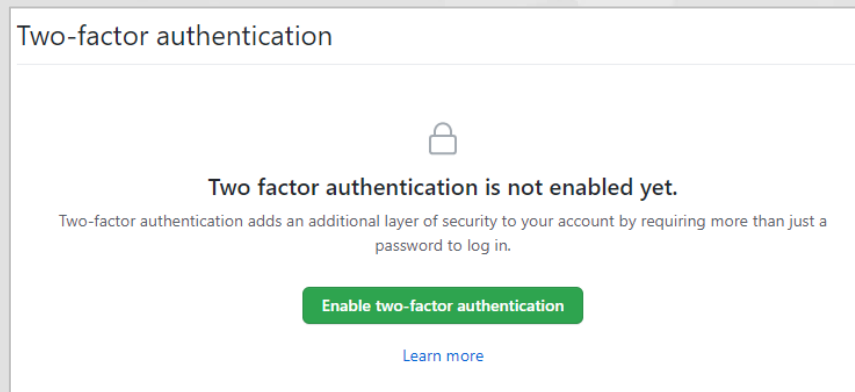
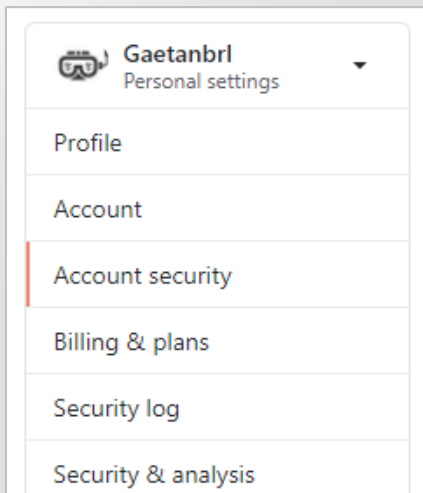
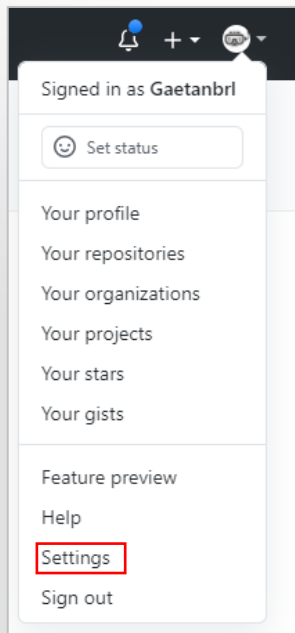
```
MINGW64:/c/Users/gaetan/Documents/git/mviewer
GNU nano 3.2

[core]
  repositoryformatversion = 0
  filemode = false
  bare = false
  logallrefupdates = true
  symlinks = false
  ignorecase = true
[remote "origin"]
  url = https://github.com/jdev-org/mviewer.git
  fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
  remote = origin
  merge = refs/heads/master
[branch "develop"]
  remote = origin
  merge = refs/heads/develop
[branch "santegraphie"]
  remote = origin
  merge = refs/heads/santegraphie
[remote "upstream"]
  url = https://github.com/geobretagne/mviewer
  fetch = +refs/heads/*:refs/remotes/upstream/*
```



### Authentication 2FA

- ≥ Pour une Organization, vous pouvez imposer une authentification à 2 facteurs à tous les utilisateurs
- ≥ Chaque compte doit alors avoir activé et géré une authentification à son niveau



## Authentication 2FA

- ≥ Il faudra choisir entre une des deux méthodes disponibles (application smartphone ou SMS)
- ≥ Dans les 2 cas, vous devrez sauvegarder les codes de récupération du compte !

### 1. Recovery codes

Recovery codes are used to access your account in the event you cannot receive two-factor authentication codes.

Download, print, or copy your recovery codes before continuing two-factor authentication setup below.

• 9a5f6-f6708	• 812ed-93617
• dfef0-31606	• d89b2-b31d5
• 08921-0b541	• 3f06f-566d1
• c071f-f9cc0	• b6bc4-2a7a4
• 92146-d60e4	• 68856-f8085
• 2aaf0-138d5	• c7912-a6d27
• 0e142-9e37d	• 78ef0-45bf9
• a32ef-30e6d	• 23d9e-394a7

Download

Print

Copy

### 2. Enable SMS two-factor authentication

We will send authentication codes to your mobile phone during sign in.

Country code

France +33

Phone number

0621303293

Authentication codes will be sent here.

Send authentication code ✓ Sent

Enter the six-digit code sent to your phone

123456

It may take a minute to arrive.

Two-factor authentication successfully enabled!

Enabled ✓ Two-factor authentication is currently on

### Génération d'un token

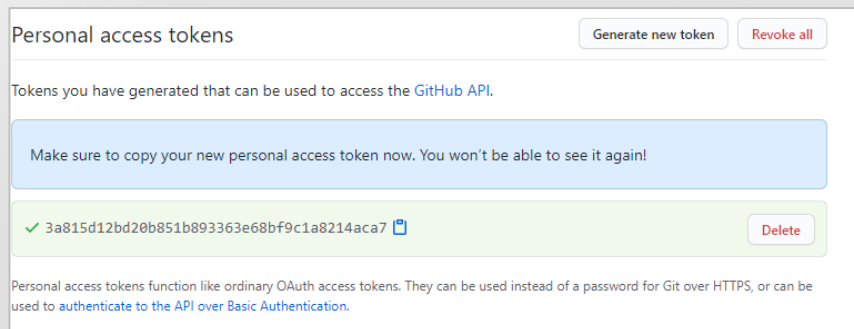
- ≥ Le token est obligatoire si une authentification 2FA est activée sur votre compte GitHub

The process of generating a GitHub Personal Access Token involves the following steps:

- Click on the profile icon in the top right corner of the GitHub interface to open the user menu.
- Click on **Settings** in the user menu.
- In the Settings page, click on **Developer settings** in the left sidebar.
- In the Developer settings page, click on **Personal access tokens** in the left sidebar.
- Click on **Generate new token** in the Personal access tokens section.
- On the "New personal access token" page, select the scopes you want to grant the token access to (e.g., **repo** for repository access, **workflow** for workflow access).
- Click on **Generate new token** at the bottom of the form.
- Copy the generated token immediately, as it will only be shown once.

## Utilisation d'un token

- ≥ Le token est à utiliser à la place de votre mot de passe au sein de la config git



## Authentication SSH (serveur)

- ≥ Une clé SSH peut également être affectée afin de sécuriser la connexion entre un serveur et votre machine



Besoin d'aller plus loin ?

<https://git-scm.com/book/fr/v2/Git-sur-le-serveur-Génération-des-clés-publiques-SSH>

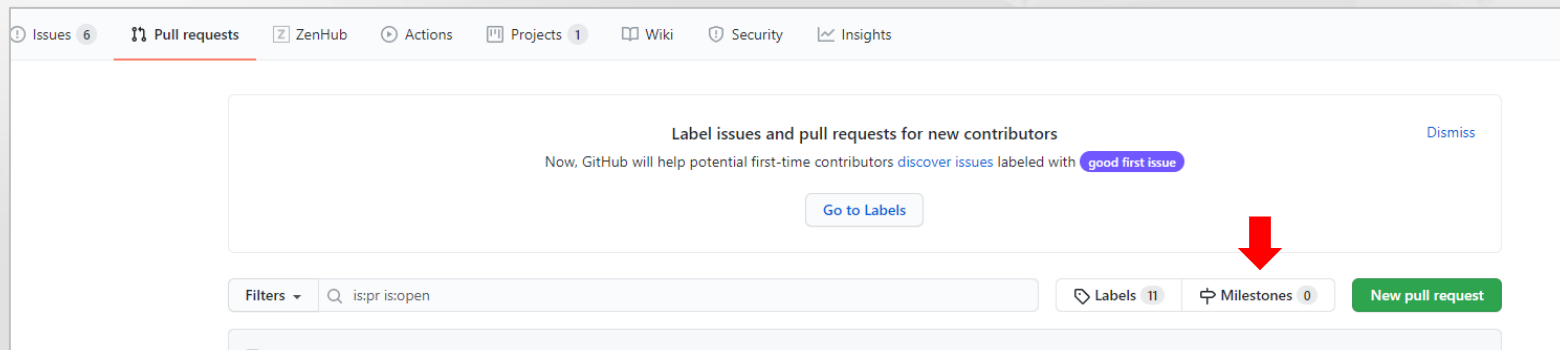
# ► **Travailler en équipe**

Release et Milestone



## Milestone

- ≥ Un Milestone est une « étape » qui est définie dans GitHub dans la feuille de route d'une version d'un projet
- ≥ Un Milestone à une durée
- ≥ Les issues et PR peuvent être rattachées à un Milestone



### Issues, PR et Milestone

- ≥ En affectant un Milestone à des issues ou à une PR, on peut suivre l'avancement de l'étape

The screenshot shows the GitHub interface for a milestone named 'v3.5'. At the top, there are tabs for 'Labels' and 'Milestones', with 'Milestones' being the active tab. A green button labeled 'New milestone' is in the top right corner. Below the tabs, a summary bar shows '1 Open' and '6 Closed' issues, with a 'Sort' dropdown on the right. The main section for the 'v3.5' milestone includes a progress bar that is 50% complete (green), indicating 7 open and 7 closed issues. Below the progress bar, it states 'No due date' and 'Last updated 6 days ago'. At the bottom of the milestone card, there are links for 'Edit', 'Close', and 'Delete'.

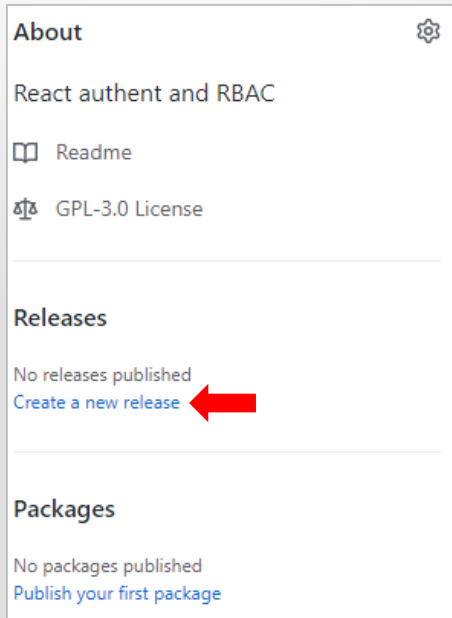
Besoin d'aller plus loin ?

<https://docs.github.com/en/free-pro-team@latest/github/managing-your-work-on-github/about-milestones>



### Release

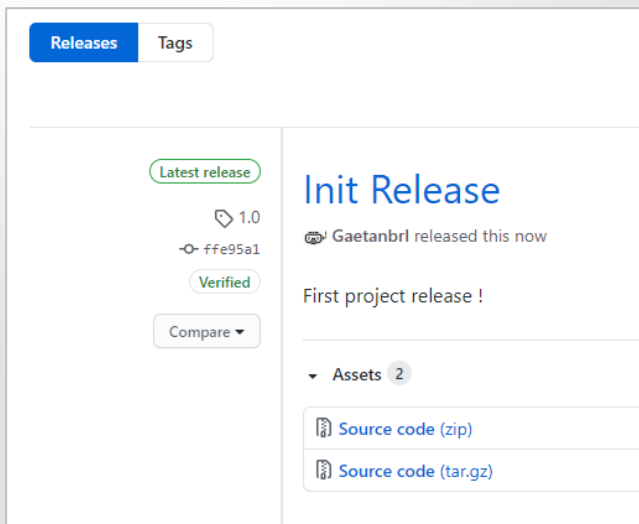
- ≥ Une release est une version du code source
- ≥ On peut intégrer des binaires



The image shows the 'Releases' form in a web application. It has tabs for 'Releases' and 'Tags'. The form includes a version input field with '1.0', a target dropdown set to 'master', and a message: 'Excellent! This tag will be created from the target when you publish this release.' Below this is an 'Init Release' button. There are 'Write' and 'Preview' tabs. The main text area contains 'First project release !'. Below the text area is a dashed box for attaching files with the instruction 'Attach files by dragging & dropping, selecting or pasting them.' and a smaller dashed box for attaching binaries with the instruction 'Attach binaries by dropping them here or selecting them.' At the bottom, there is a checkbox for 'This is a pre-release' and two buttons: 'Publish release' and 'Save draft'.

### Release

- ≥ Chaque release a un tag strictement unique et selon une logique
- ≥ La release propose alors le code source comme une archive



#### Bonnes pratiques

- Présenter les nouveautés
- Lister les correctifs apportés
- Rattacher des exemples
- Lier une documentation

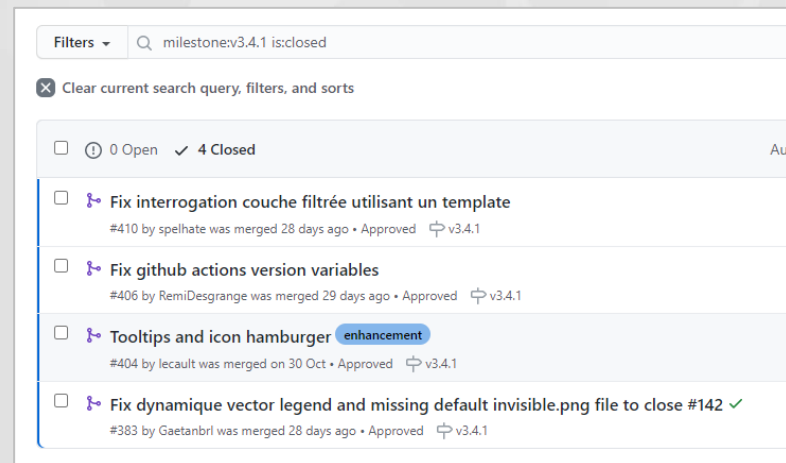
Exemples : <https://github.com/geobretagne/mviewer/releases>

## Release et Milestone

≥ Une release peut directement référencer le ou les Milestones concernés à la place de référencer les issues et correctifs en liste



The screenshot shows the GitHub release page for version v3.4.1, dated November 2020. The release is marked as the 'Latest release' and is 'Verified'. It was released by 'spelhate' 27 days ago. The release description states it is a 'Version mineure incluant 3 bug fixes.' and provides a link to the documentation: <https://mviewerdoc.readthedocs.io/fr/v3.4/>. It also mentions that users will find 'ici' the set of modifications for this version. Under the 'Assets' section, there are two links: 'Source code (zip)' and 'Source code (tar.gz)'.



The screenshot shows a GitHub issues page filtered by the milestone 'v3.4.1 is:closed'. The search query is 'milestone:v3.4.1 is:closed'. The page shows 0 Open and 4 Closed issues. The issues listed are:

- #410: Fix interrogation couche filtrée utilisant un template (by spelhate, merged 28 days ago, Approved, v3.4.1)
- #406: Fix github actions version variables (by RemiDesgrange, merged 29 days ago, Approved, v3.4.1)
- #404: Tooltips and icon hamburger (enhancement, by lecault, merged on 30 Oct, Approved, v3.4.1)
- #383: Fix dynamique vector legend and missing default invisible.png file to close #142 (by Gaetanbri, merged 28 days ago, Approved, v3.4.1)

Exemple : <https://github.com/geobretagne/mviewer/issues?q=milestone%3Av3.4.1+is%3Aclosed>

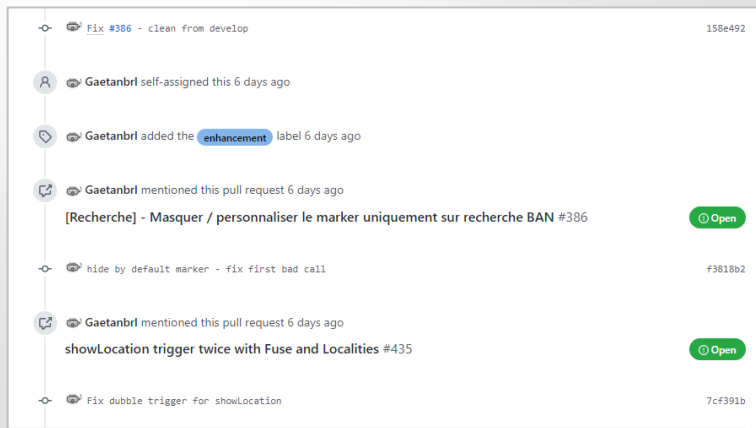
# ► **Travailler en équipe**

Autolink



### Qu'est-ce que c'est ?

- ≥ Les références permettent de référencer automatiquement une issue, une PR ou un compte dans un texte
- ≥ Ces références sont directement converties en URL comme un raccourci



Besoin d'aller plus loin ?

<https://docs.github.com/en/free-pro-team@latest/github/writing-on-github/autolinked-references-and-urls>



