# Présentation de Git et Github

Rappel des commandes de base <https://git-scm.com/book/fr/v1/Les-bases-de-Git>

<https://gist.github.com/aquelito/8596717>

# Utilisation des commandes Fork, Pull-request, Branch

# Fork

Par défaut, il n'est pas possible de faire de commit sur un dépôt qui ne nous appartient pas. On utilise donc « **fork**» pour avoir une copie du dépôt ce qui permet de se retrouver avec un dépôt sur lequel on aura la permission d'écriture.

# Pull request

Une « **pull request** » consiste tout simplement à demander à l'auteur du dépôt original de prendre en compte les modifications que vous avez apportées sur votre fork et que vous souhaitez partager

Plusieurs possibilités s’offre alors à lui :

* commenter votre demande (par exemple si votre code ne respecte pas les normes)
* refuser votre demande ou l’accepter c’est-à-dire faire un **merge**

# Branch

Pour un projet commun, travailler sur plusieurs branches s’avère utile lorsque l’on souhaite tester une partie du projet, ou encore se concentrer sur le développement d'une fonctionnalité spécifique. Par défaut lorsqu’on initialise un dépôt Git, notre code est placé dans la branche principale  **master.**

Pour voir toutes les branches présentes dans le dépôt, on utilise la commande **git branch**. Elle retourne les branches présentes et indique la branche sur laquelle l’on est placé.

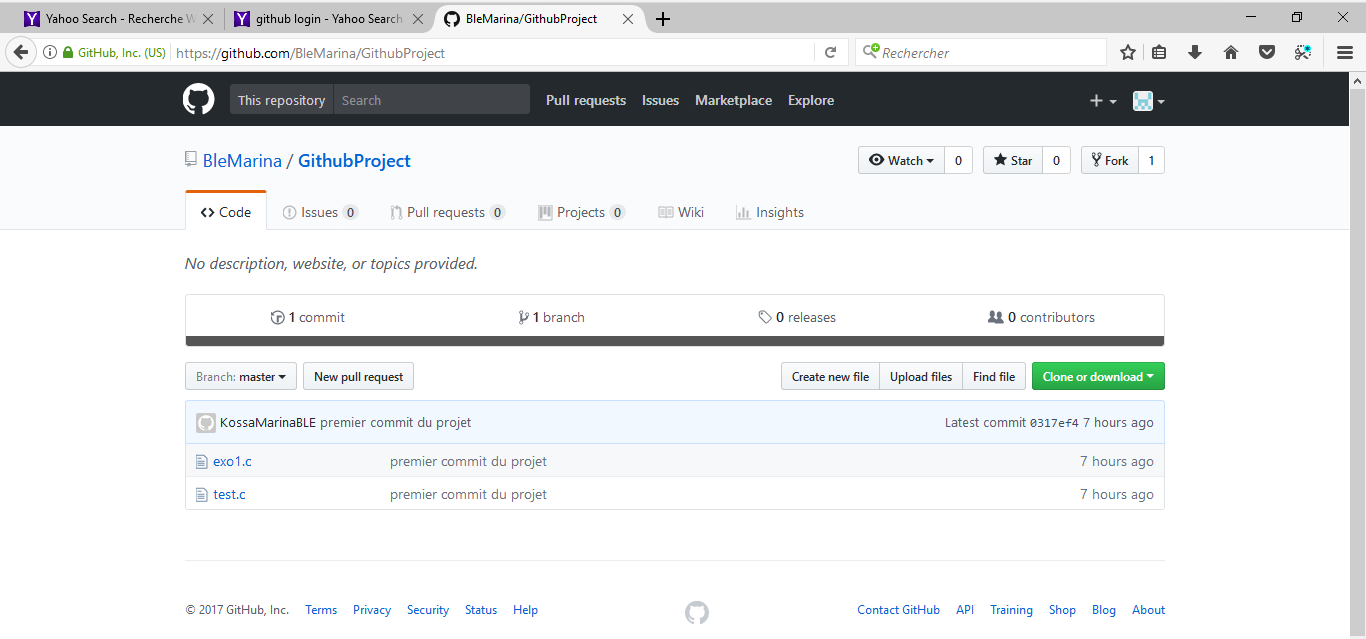
* **git branch «nom\_branche»** pour créer une nouvelle branche
* **git checkout «nom\_branche»** permet de se placer sur une autre branche à l'intérieur de notre repository

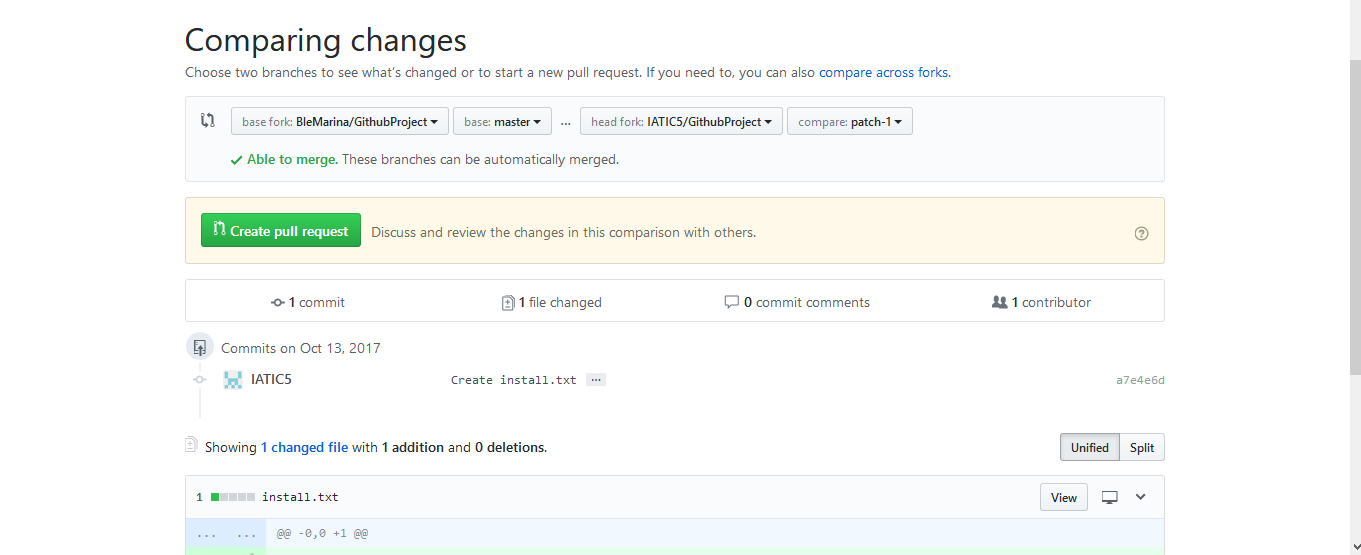
On peut regrouper ces deux commandes en une seule

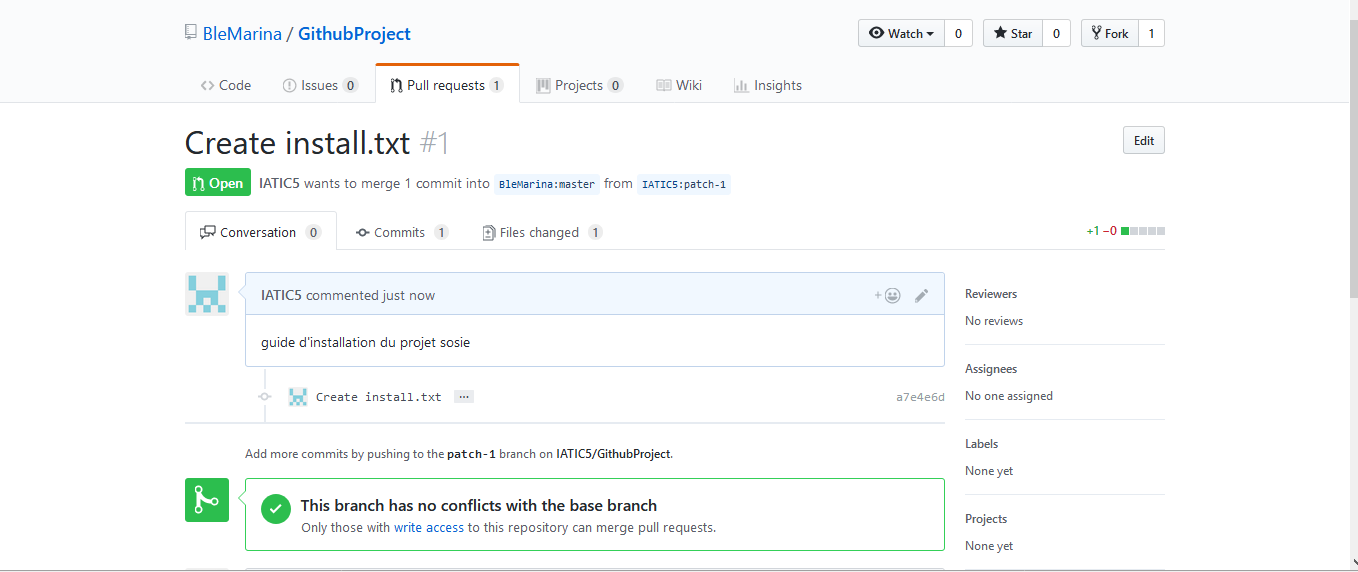
* **git checkout –b «nom\_branche»** pour créer une branche et s’y positionner

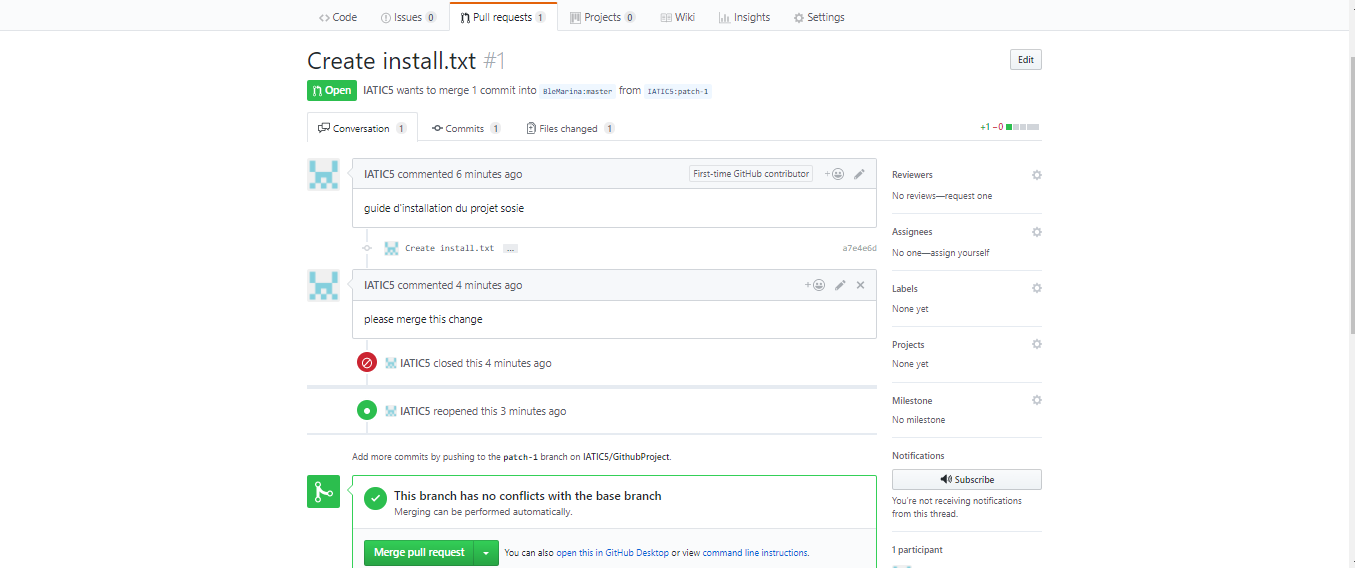
# Fonctionnement sur Github et marche à suivre en passant par l’interface graphique de Github

Pour commencer, On créé un new Repository pour y déposer les fichiers exo1.c et test.c

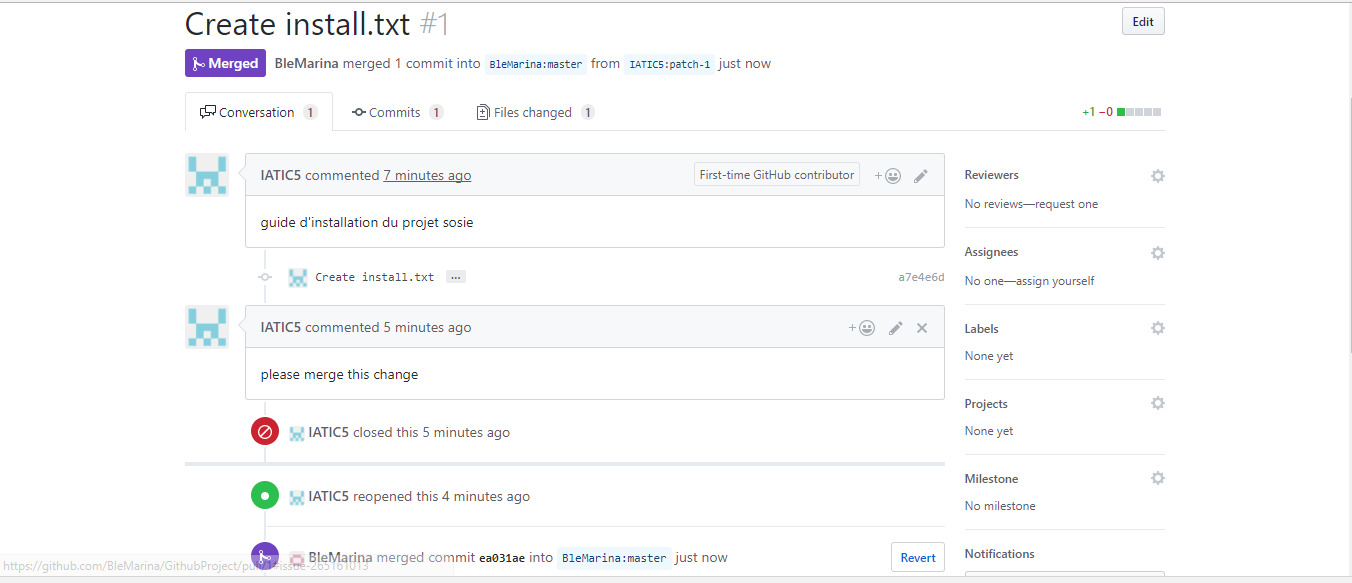
On créé également un autre utilisateur qui souhaite avoir une copie du dépôt en local afin de pourvoir y apporter des modifications. Pour ce faire, on fait un « Fork »

Ce utilisateur créé un nouveau fichier install.txt ( **create new file**)et demande une « pull request » (**create pull request**) sur la branche principale.

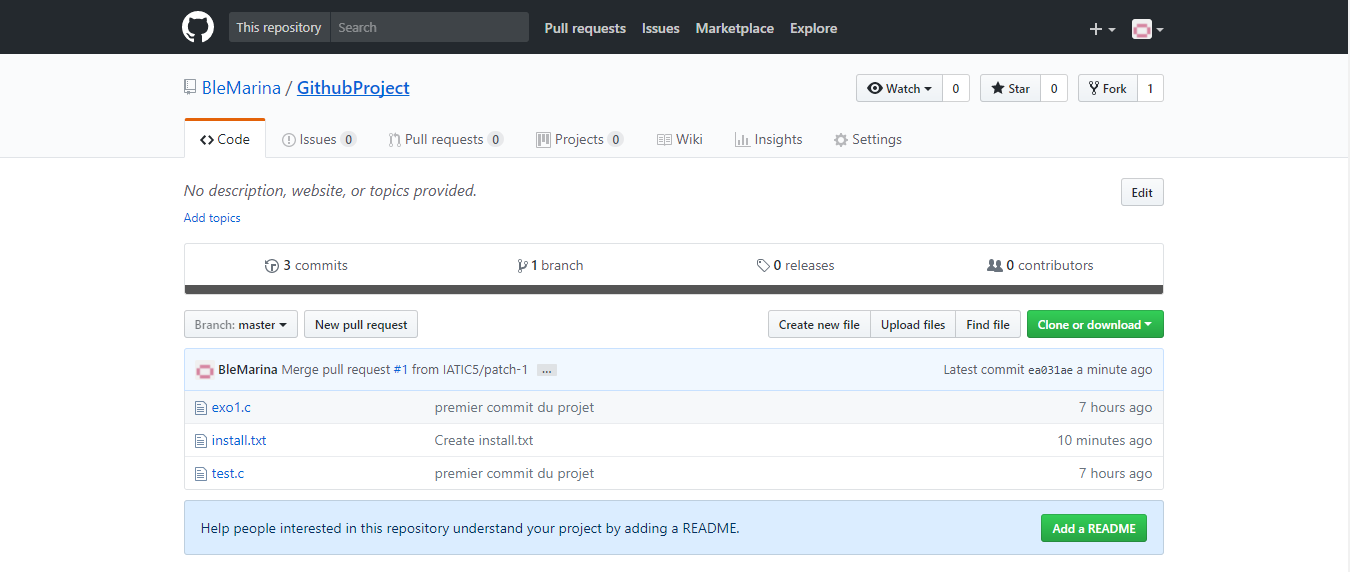
On vérifie qu’il n’y a pas de conflits de merge entre sa branche locale et la branche principale.



On voit que ce n’est pas le cas , le merge peut être valider



Les modifications sont bien visibles dans la branche principale



## Quelques lignes de commandes pour effectuer une pull-request

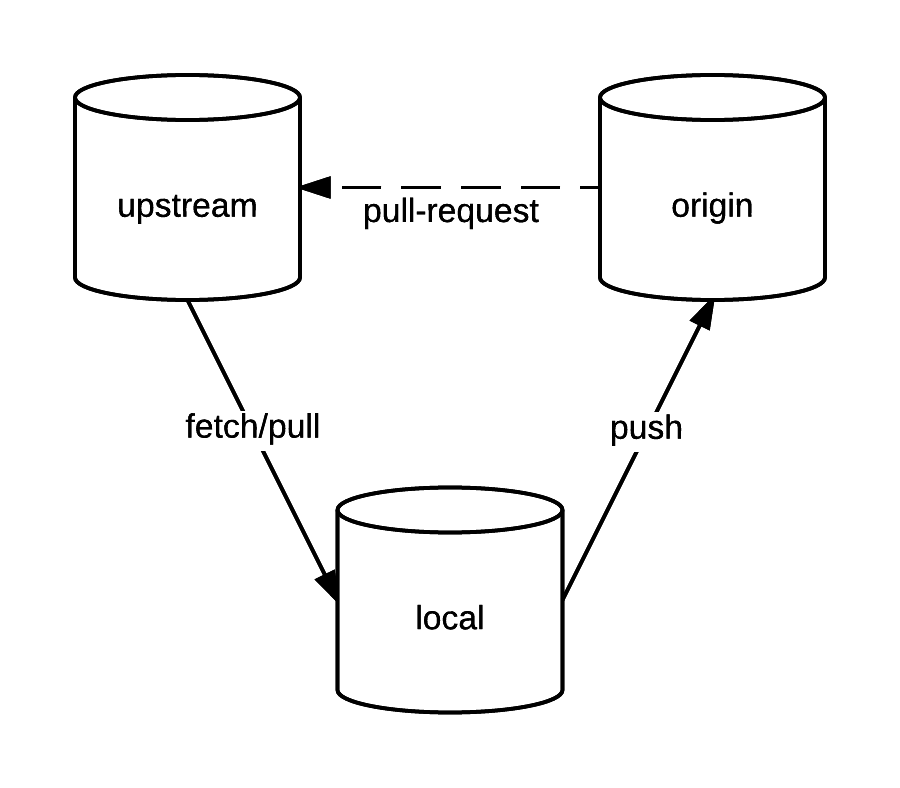


Figure 1: Schéma de principe du pull request

Pour commencer, faire un fork du dépôt original sur le GUI de Github

Il faut que notre fork soit sur notre machine, pour cela on fait un clone :

|  |
| --- |
| **git clone [url]** |

Nous travaillerons sur notre dépôt local en face du fork

Ensuite, nous allons créer une branche dédiée à la tâche que nous avons effectuée et nous positionner dessus. Par exemple si l’on souhaite faire des améliorations, on crée une branche « brancheTest » Cette branche sera une copie de la branche master (pour être bien à jour avec le master) :

|  |
| --- |
| **git checkout –b brancheTest master** |

Nous pouvons désormais faire toutes les modifications que nous souhaitons et écrire nos commits.

Pour effectuer un push ,il va falloir que notre dépôt soit synchro avec le dépôt d'origine. Pour cela Nous allons donc ajouter à notre fork une liaison avec le dépôt original :

|  |
| --- |
| **git remote add upstream  [url-du-fork]** |

Ensuite, mettez-nous sur notre branche master et récupérez les modifications de la branche master du dépôt original pour être synchro :

|  |
| --- |
| **git checkout master git pull --ff-only upstream master** |

On se repositionne sur notre branche locale :

|  |
| --- |
| **git checkout brancheTest** |

Pour garder un historique de branche propre, on fait un rebase.

Avec le rebase, nos commit seront réappliqués à la suite de votre branche master (comme si nous n'aviez travaillé que sur une seule branche) :

|  |
| --- |
| **git rebase master** |

Il ne nous reste plus qu'à pousser nos modifications en ligne :

|  |
| --- |
| **git push origin brancheTest** |

Il se peut que Git nous demandera de faire un git pull, pour éviter les conflits , nous allons forcez l'envoi des modifications en faisant

|  |
| --- |
| **git push -f origin brancheTest** |

http://thelia-school.com/media/upload/faire-pull-request/pull-request-button.pngNos modifications sont maintenant en ligne mais pour le moment, le dépôt d'origine n'est pas touché. Il faut maintenant demander la fusion de nos modifications sur le projet. Pour ce faire, on se rend sur le GUI de Github