

Git

Git branch = permet de voir toutes les branches du projet.

```
dorch@DESKTOP-STLUBST MINGW64 ~/Desktop/PROJETS/PERSO/Git-Apprentissage (master)
$ git branch
  contact-page
  footer
  header
* master
```

Git branch feature/nom = permet de créer une branche.

!! Toujours se positionner sur la racine du projet pour créer une nouvelle branche (master ou main). Sinon, cela peut engendrer des conflits.

```
dorch@DESKTOP-STLUBST MINGW64 ~/Desktop/PROJETS/PERSO/Git-Apprentissage (master)
$ git branch form/payne

dorch@DESKTOP-STLUBST MINGW64 ~/Desktop/PROJETS/PERSO/Git-Apprentissage (master)
$ git branch
  contact-page
  footer
  form/payne
  header
* master
```

Git checkout nomdelabranche = permet de switcher de branche en branche.

```
dorch@DESKTOP-STLUBST MINGW64 ~/Desktop/PROJETS/PERSO/Git-Apprentissage (master)
$ git checkout form/payne
Switched to branch 'form/payne'

dorch@DESKTOP-STLUBST MINGW64 ~/Desktop/PROJETS/PERSO/Git-Apprentissage (form/payne)
$ |
```

Tapez **code** . pour ouvrir plus rapidement votre projet sur VS Code. Travaillez sur votre branche normalement.

Faire `git add .` et `git commit -m «commentaire»` comme d'habitude.

Git push origin nomdelabranche = permet de faire un push sur la branche adéquate.

```
dorch@DESKTOP-STLUBST MINGW64 ~/Desktop/PROJETS/PERSO/Git-Apprentissage (form/payne)
$ git add .

dorch@DESKTOP-STLUBST MINGW64 ~/Desktop/PROJETS/PERSO/Git-Apprentissage (form/payne)
$ git commit -m "Form init"
[form/payne 8c4ab51] Form init
1 file changed, 1 insertion(+)

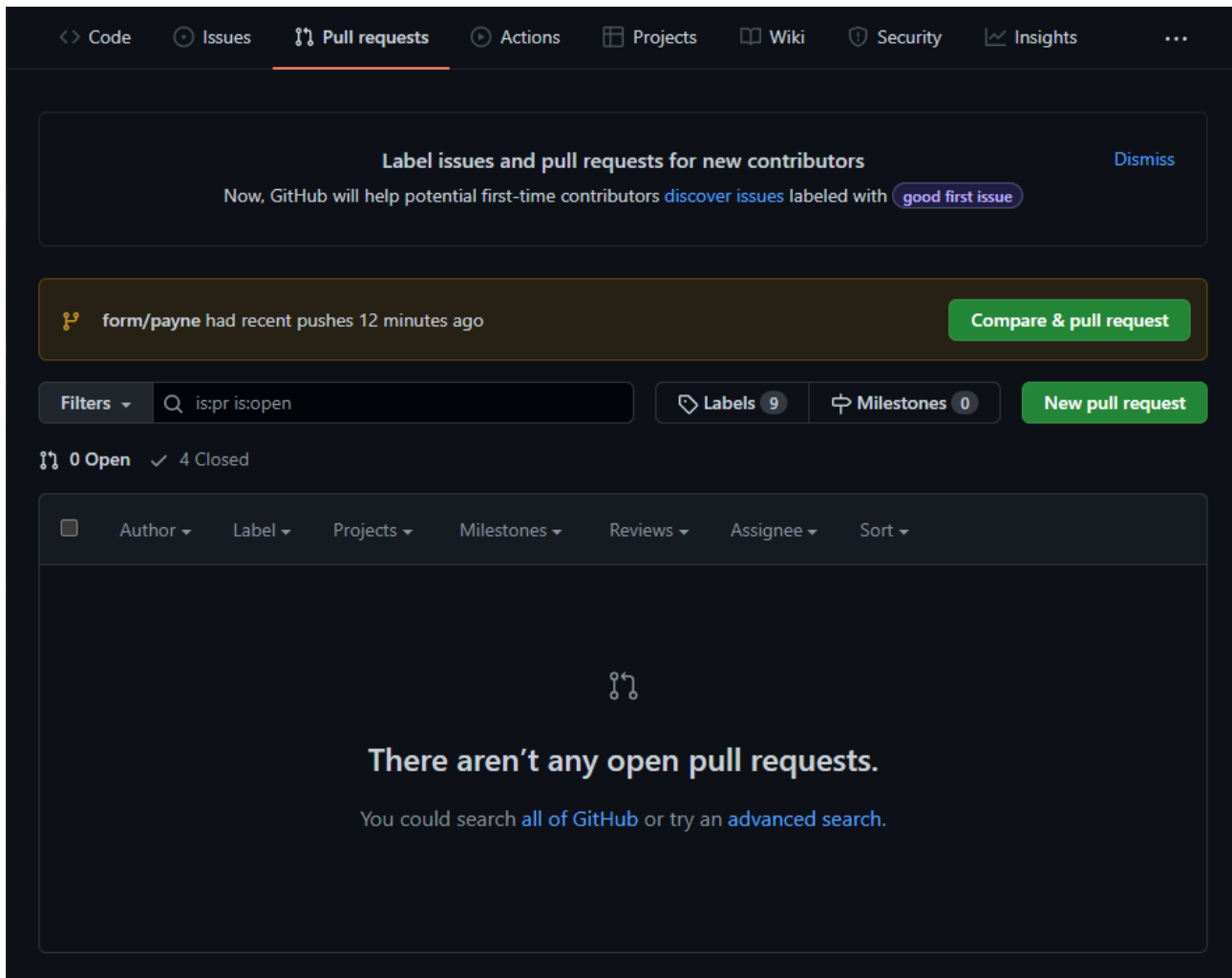
dorch@DESKTOP-STLUBST MINGW64 ~/Desktop/PROJETS/PERSO/Git-Apprentissage (form/payne)
$ git push origin form/payne
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 279 bytes | 279.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'form/payne' on GitHub by visiting:
remote:   https://github.com/CaroD59/Git-Apprentissage/pull/new/form/payne
remote:
To https://github.com/CaroD59/Git-Apprentissage.git
 * [new branch]      form/payne -> form/payne
```

Pour que les collègues puissent récupérer le travail :

- Se positionner sur la racine (master ou main).
- Taper `git pull` tout simplement
- Bien communiquer et se coordonner avec ses collègues pour le nom des branches !
- Taper `git checkout nomdelabranche` (que vous souhaitez récupérer).
- Se remettre sur la racine (master ou main)
- Retaper un petit `git branch` pour voir si toutes les branches ont bien été récupéré.

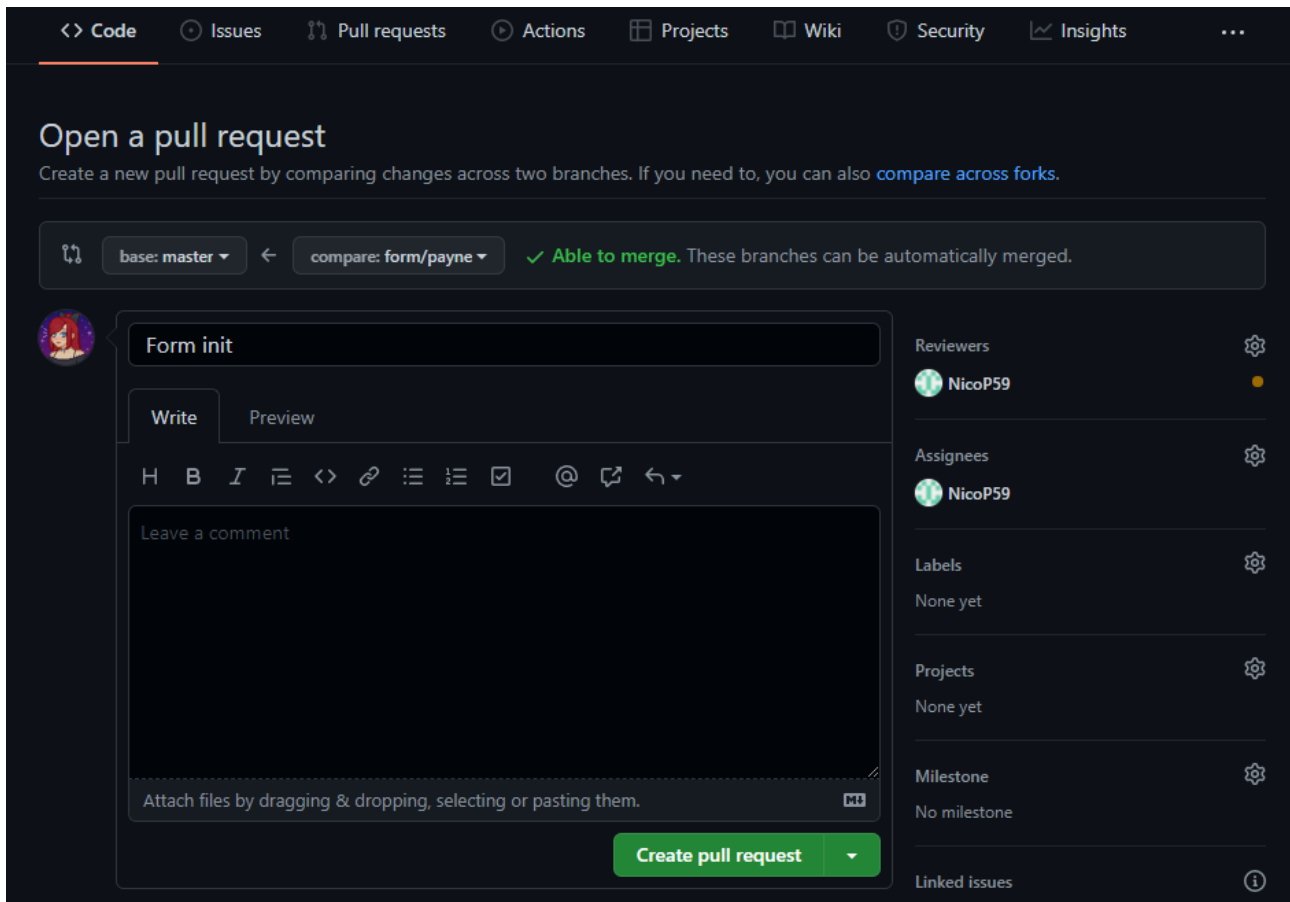
PULL REQUEST

Pull request ou PR (onglet sur votre projet Github) permet de **merger** (combiner) vos branches sur la master ou main.



Ici, nous pouvons ouvrir une pull request. C'est une requête que l'on fait à notre chef de projet ou product owner qui s'occupe de relire et vérifier votre code de la branche adéquate avant de vous donner l'autorisation de la combiner sur la master/main.

Tapez sur **Compare & Pull Request** de la branche souhaitée.



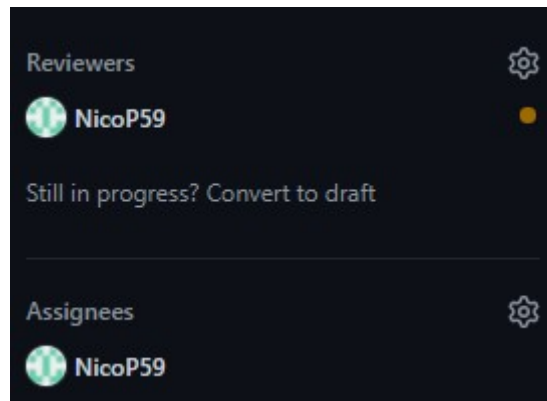
✓ **Able to merge** signifie qu'il n'y a aucun conflit entre cette branche et la master/main. En **rouge**, il y aura bien évidemment des conflits à résoudre...



Ici, sur la **master**, on souhaite merger la branche **form/payne**.

Créer des commentaires professionnels, simples et nets en rapport avec ce que vous avez faits pour que votre chef de

projet et vos collègues puissent savoir où vous en êtes concrètement.



A droite de l'écran :

- **Les reviewers** : taguez vos collègues pour qu'ils puissent vérifier votre code avec vous. On est jamais de trop pour vérifier si on a pas fait d'erreurs.
- **Les assignees** : taguez votre chef de projet ou product owner pour qu'il puisse valider votre code avant de merger.

Appuyer sur **Create Pull Request**.

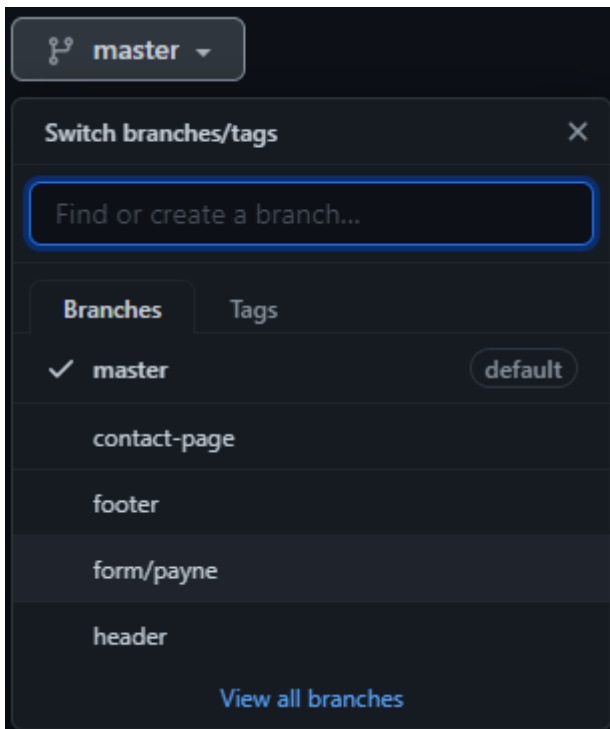
The screenshot shows a GitHub Pull Request (PR) titled "Form init #5". At the top, it indicates "CaroD59 wants to merge 1 commit into master from form/payne". Below this, there are tabs for Conversation (0), Commits (1), Checks (0), and Files changed (1). The main content area shows a comment from CaroD59: "No description provided." Below the comment, there is a commit titled "Form init" with hash 8c4ab51. Further down, it shows that CaroD59 requested a review from NicoP59 and assigned NicoP59. On the right side, there are sections for Reviewers (NicoP59), Assignees (NicoP59), Labels (None yet), Projects (None yet), Milestone (No milestone), and Linked issues (None yet). At the bottom, there is a green box with a checkmark indicating "Review requested" and "This branch has no conflicts with the base branch". A green button labeled "Merge pull request" is visible.

Votre PR est créée, **Form init #5** est le nom de la pull request. On retrouve un descriptif de tout ce qu'il a été créé. Les reviewers et assignees pourront valider que la branche ne contient bien aucun conflit et qu'elle est prête à être mergée.

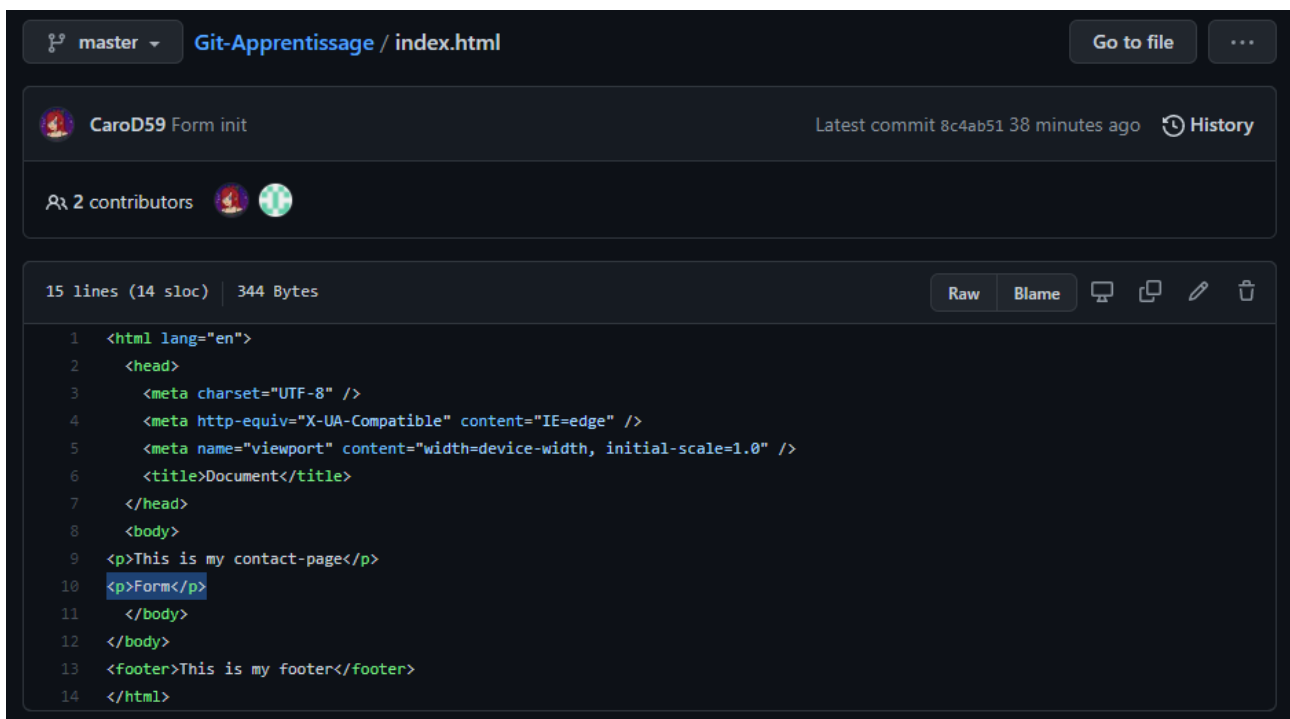
(Ici, étant un trop petit projet juste pour présenter les fonctionnalités cela reste très restreint.)

Aucun conflit ? Tout est vérifié ? Il est temps d'appuyer sur **Merge Pull Request** et **Confirm Merge** après avoir commenté correctement votre travail !

The screenshot shows a GitHub notification with a purple icon of a pull request. The text reads: "Pull request successfully merged and closed. You're all set—the form/payne branch can be safely deleted." There is a button labeled "Delete branch" on the right.



On retrouve bien ici notre branche **form/payne**.



Nous pouvons voir ici que le travail de la branche **form/payne** a bien été mergée.

Les collègues peuvent récupérer la partie de leur camarade en allant sur la **branche racine du projet** (master/main) et en faisant un petit **git pull** 😊