

Workflow with GitHub Desktop



Installer [GitHub Desktop](#) pour la gestion de Git et [Visual Studio Code](#) pour la gestion de conflits.

1. "Open in Desktop" pour cloner le projet à votre l'ordinateur



The screenshot shows the GitHub repository page for 'NicholasLangevin / Guide_de_survie_en_actuariat'. The repository is private and has 104 commits, 9 branches, 0 releases, 3 contributors, and is licensed under GPL-3.0. The description is 'The only document you need to be an Actuary'. There are tags for 'actuarial-science', 'university', and 'latex-document'. A recently pushed branch 'alec_modif' is shown. A modal for cloning the repository is open, showing the 'Clone with HTTPS' option and the URL 'https://github.com/NicholasLangevin/Guid'. The 'Open in Desktop' button is highlighted with a blue box.


2. Une fois le projet cloné, créer une nouvelle branche.

The screenshot shows the GitHub Desktop interface. The 'Current Branch' is 'alec_modif'. The 'Fetch origin' button is visible. The 'Branches' tab is selected, showing a list of branches. The 'New Branch' button is highlighted with a blue box. The 'Default Branch' is 'master', and the 'Recent Branches' list includes 'alec_modif'.

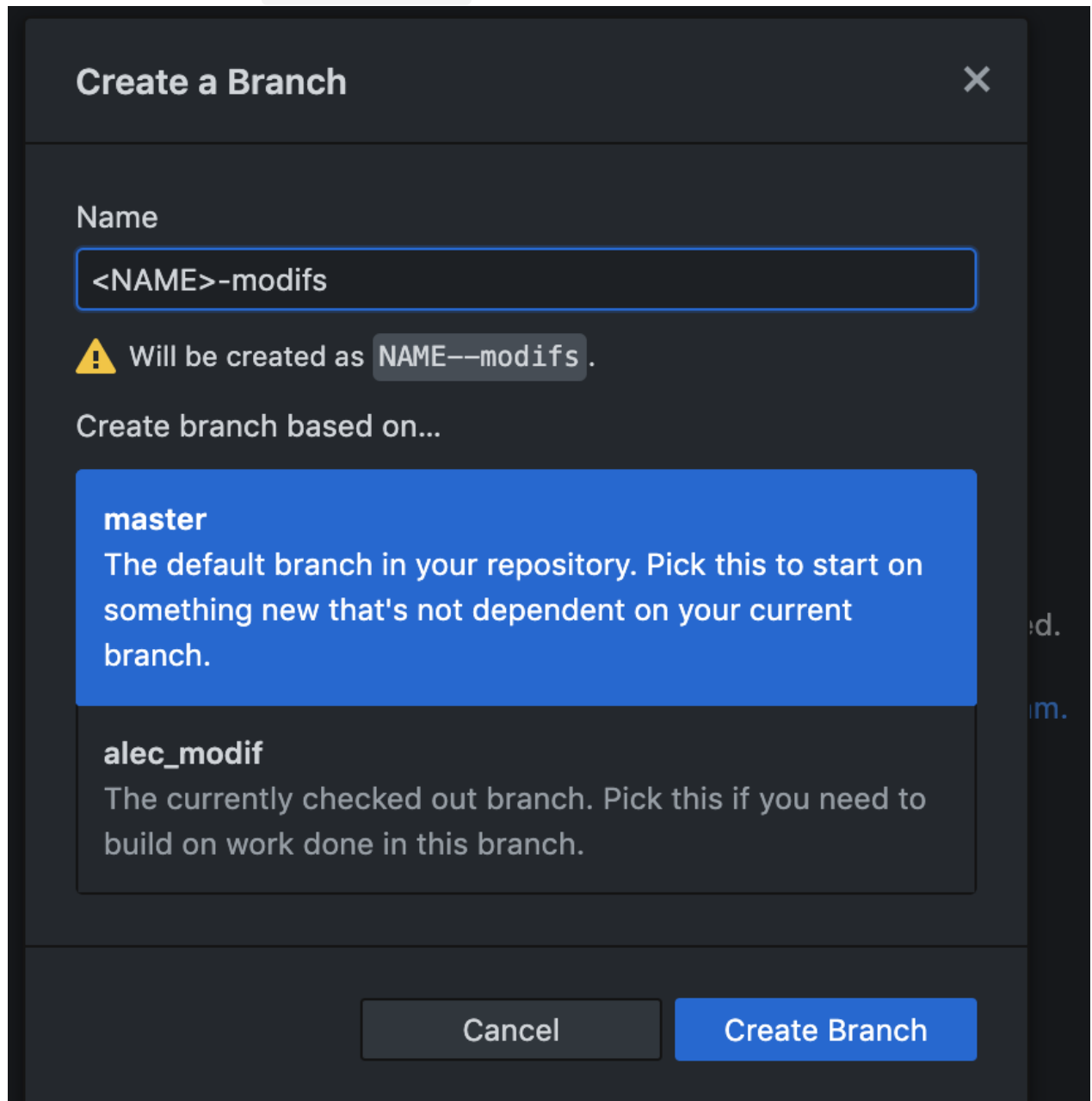
-  ajout_intro1 7 months ago
-  app_flashcard a month ago

Other Branches

-  origin/ajout_gitworkflow 7 months ago
-  origin/fisher_partiel 7 months ago
-  origin/make a month ago
-  origin/modif_gab 2 months ago
-  origin/modif_typo 7 months ago

 Choose a branch to merge into **alec_modif**

3. Nommer la branche `<PRÉNOM>-MODIFS`



The screenshot shows a 'Create a Branch' dialog box with a dark theme. At the top, the title 'Create a Branch' is on the left and a close button 'X' is on the right. Below the title, the 'Name' field contains the text '<NAME>-modifs'. A warning icon (yellow triangle with an exclamation mark) is followed by the text 'Will be created as NAME--modifs.'. Underneath, the text 'Create branch based on...' is followed by two options: 'master' (highlighted in blue) and 'alec_modif'. The 'master' option has a description: 'The default branch in your repository. Pick this to start on something new that's not dependent on your current branch.' The 'alec_modif' option has a description: 'The currently checked out branch. Pick this if you need to build on work done in this branch.' At the bottom, there are two buttons: 'Cancel' and 'Create Branch'.

Create a Branch X

Name

<NAME>-modifs

⚠ Will be created as NAME--modifs.

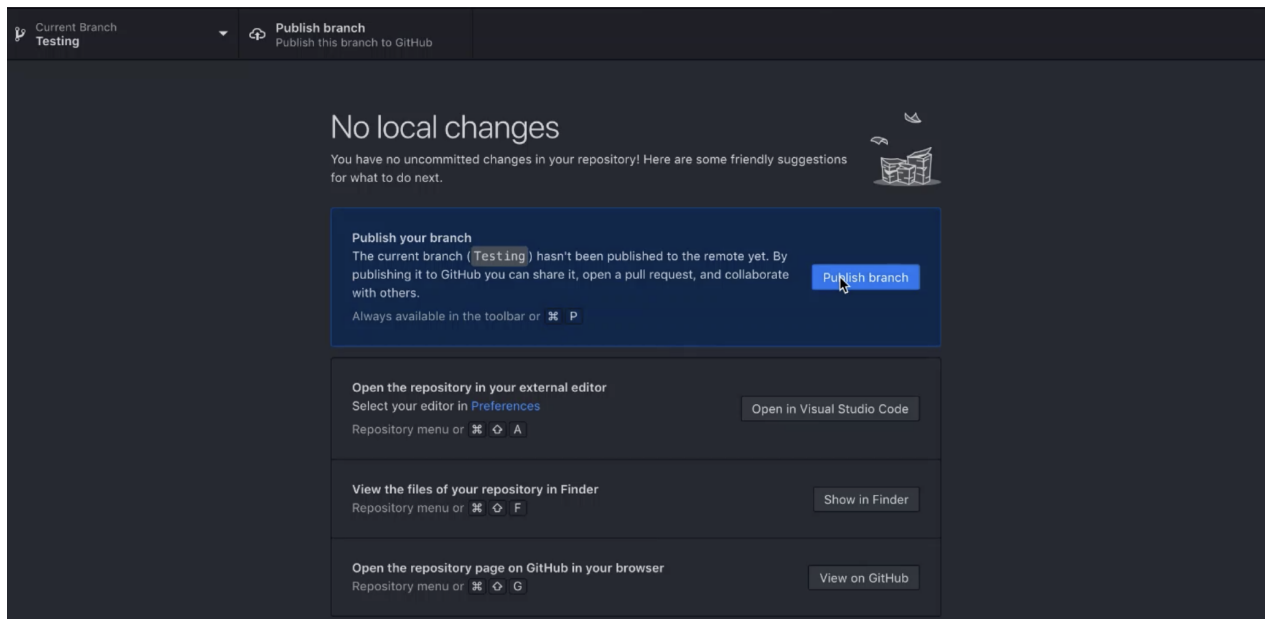
Create branch based on...

master
The default branch in your repository. Pick this to start on something new that's not dependent on your current branch.

alec_modif
The currently checked out branch. Pick this if you need to build on work done in this branch.

Cancel Create Branch

4. Publier votre branche!



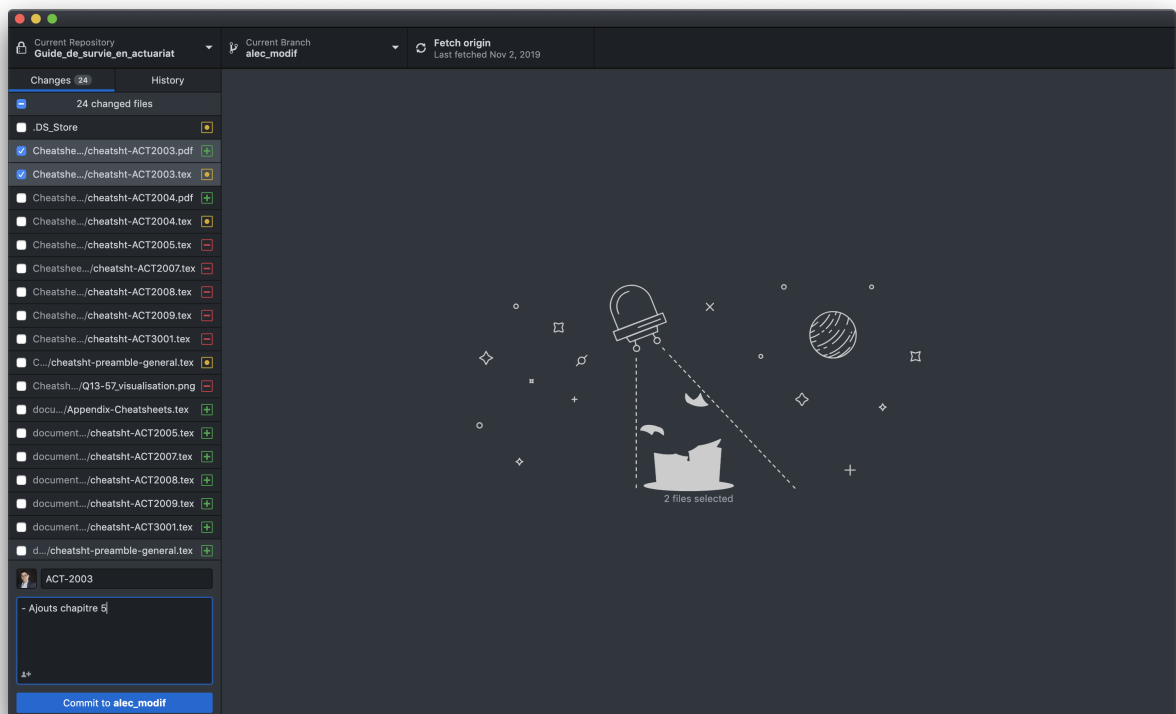
Vous pouvez maintenant effectuer des modifications!

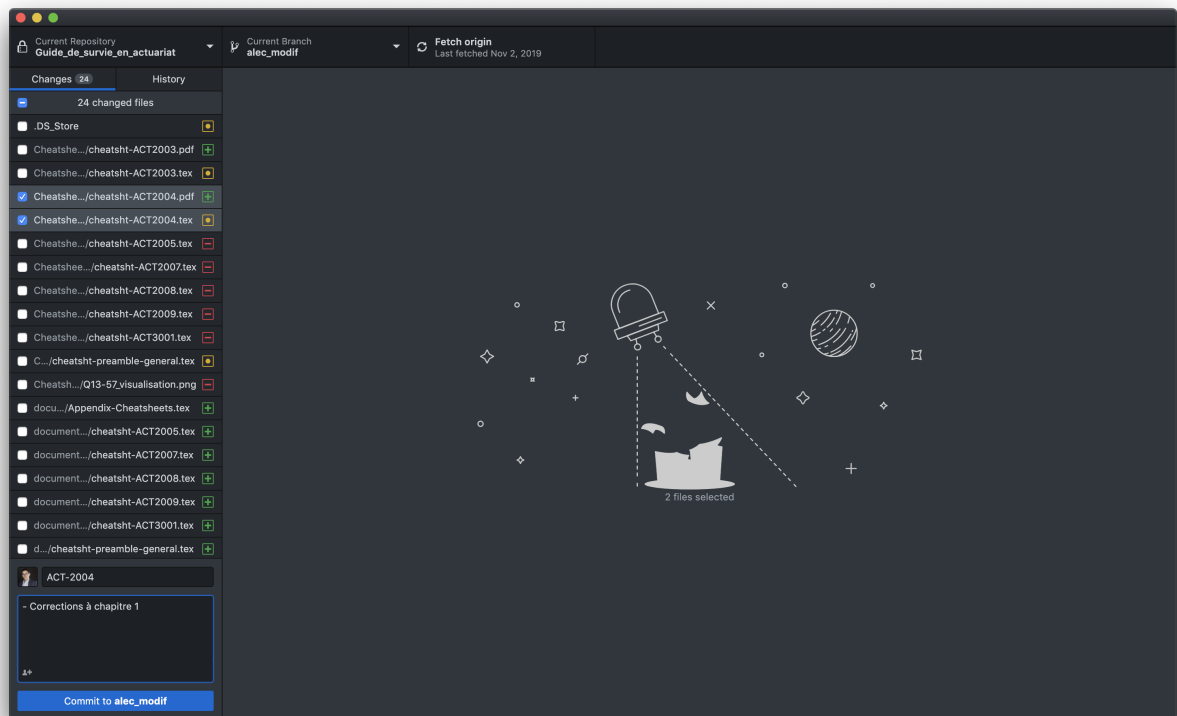
Contribuer

ASSUREZ-VOUS D'ÊTRE DANS VOTRE BRANCHE

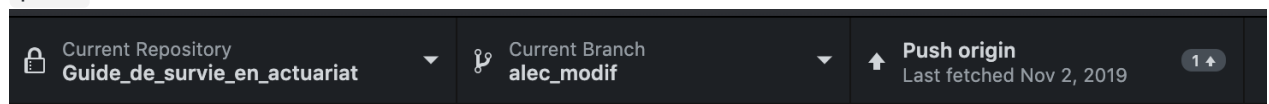
1. Garder vos `commit` simples pour qu'ils soient facilement acceptables.

Par exemple, lorsque vous effectuez des modifications faites un `commit` par cours modifié.





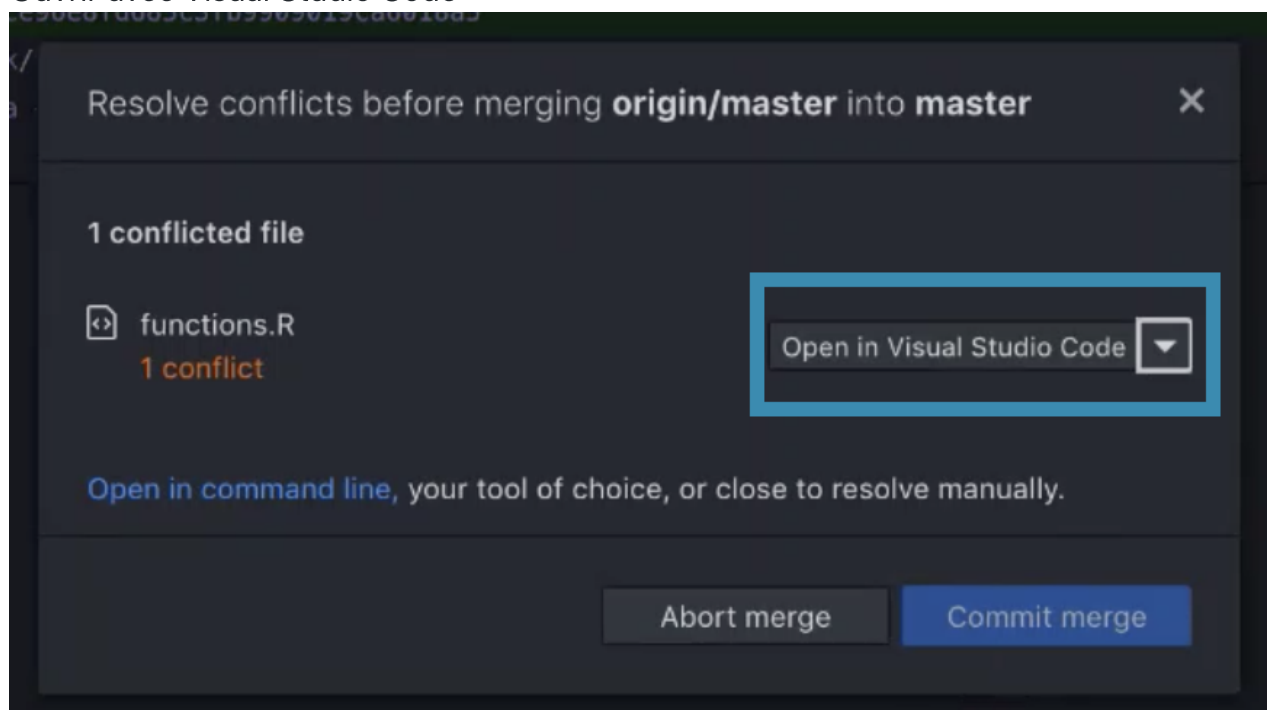
2. push !



Gérer les conflits

Lorsque vous avez cette erreur, c'est qu'il y a un conflit; n'ayez pas peur, Visual Studio Code permet de la gérer visuellement.

1. Ouvrir avec Visual Studio Code



2. Défiler jusqu'à ce que vous trouvez une partie du code ressemblant à ceci:

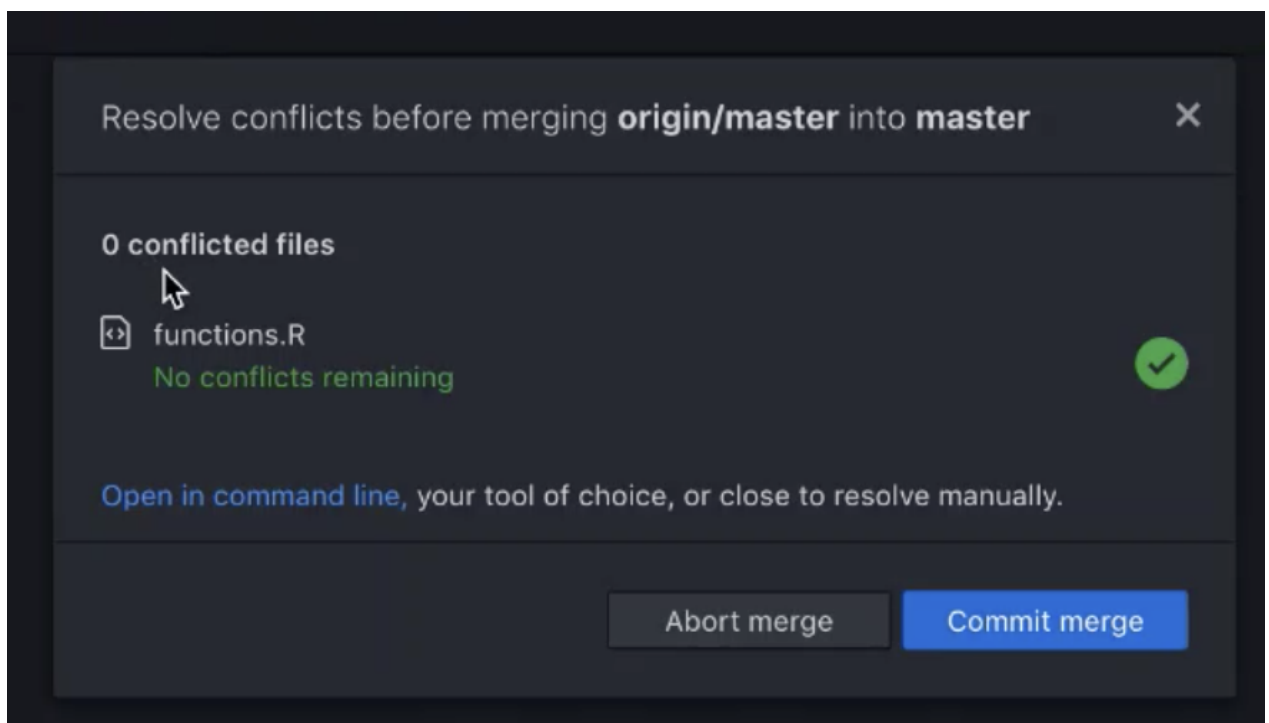
```
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<< HEAD (Current Change)
(1 / gamma(alpha)) * lam^(k / tau) *
=====
(1/gamma(alpha)) *
lam^(k/tau) *
>>>>>> 608c5c28b4ce96e8fd685c3fb9909019ca6018a5 (Incoming Change)
gamma(1 + k/tau) *
gamma(alpha - k/tau)
```

Suffit de choisir quelle modification conserver!

Suite au choix, sauvegarder et retourner dans GitHub Desktop.

3. Le message suivant va apparaître.

On peut donc faire le `commit` et faire un push de nos modifications.



4. Vous pouvez maintenant faire un `push` de vos modifications

Rédiger un `Pull Request` sur Github

- On rédige notre `pull request` directement sur le site du serveur du dépôt.
 - L'interface web nous indique en temps réel si des merge conflits on lieu.
- Attribuer (`assign`) la vérification de votre requête à un des collaborateurs en charge du projet pour qu'il accepte la modification.
 - Il impératif d'assigner un contributeur étant présentement dans le cours ou l'ayant déjà complété.
- Assigner une libellée (`label`) pertinente à votre ajout.

Exemples

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master

compare: alec_modif

✓ Able to merge.

These branches can be automatically merged.

ACT-2003

Write

Preview

AA

B

i

“

<>

↻

⋮

⋮

⋮

@

🔖

↶

- Ajouts chapitre 5

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Reviewers

Suggestions

gabrielcrepeault

Request

Assignees

No one—assign yourself

Labels

Proposition d'ajout

Projects

None yet

Milestone

No milestone

1 commit

2 files changed

0 commit comments

1 contributor

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master

compare: alec_modif

✓ Able to merge.

These branches can be automatically merged.

ACT-2004

Write

Preview

AA

B

i

“

<>

↻

⋮

⋮

⋮

@

🔖

↶

- Corrections à chapitre 1

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Reviewers

Suggestions

gabrielcrepeault

Request

Assignees

No one—assign yourself

Labels

Typo

Projects

None yet

Milestone

No milestone

1 commit

2 files changed

0 commit comments

1 contributor