Merging Branches and Git Flow

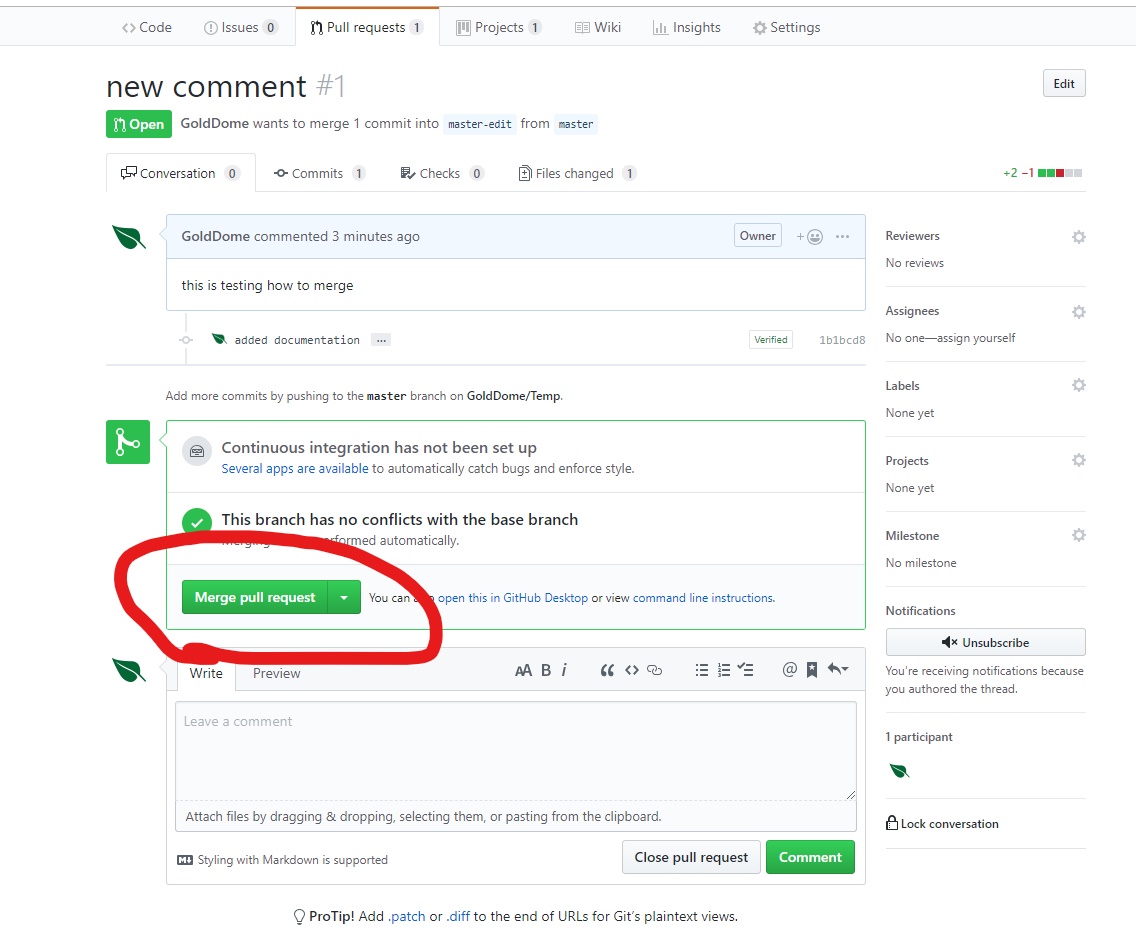
Merging

The final step of editing any repository is to merge two branches. One of the branches will always be the master branch and the other will be a copy of the master branch that has been edited. By merging the two branches programmers can take the changes made in one branch and apply them to the other. Both branches will remain.

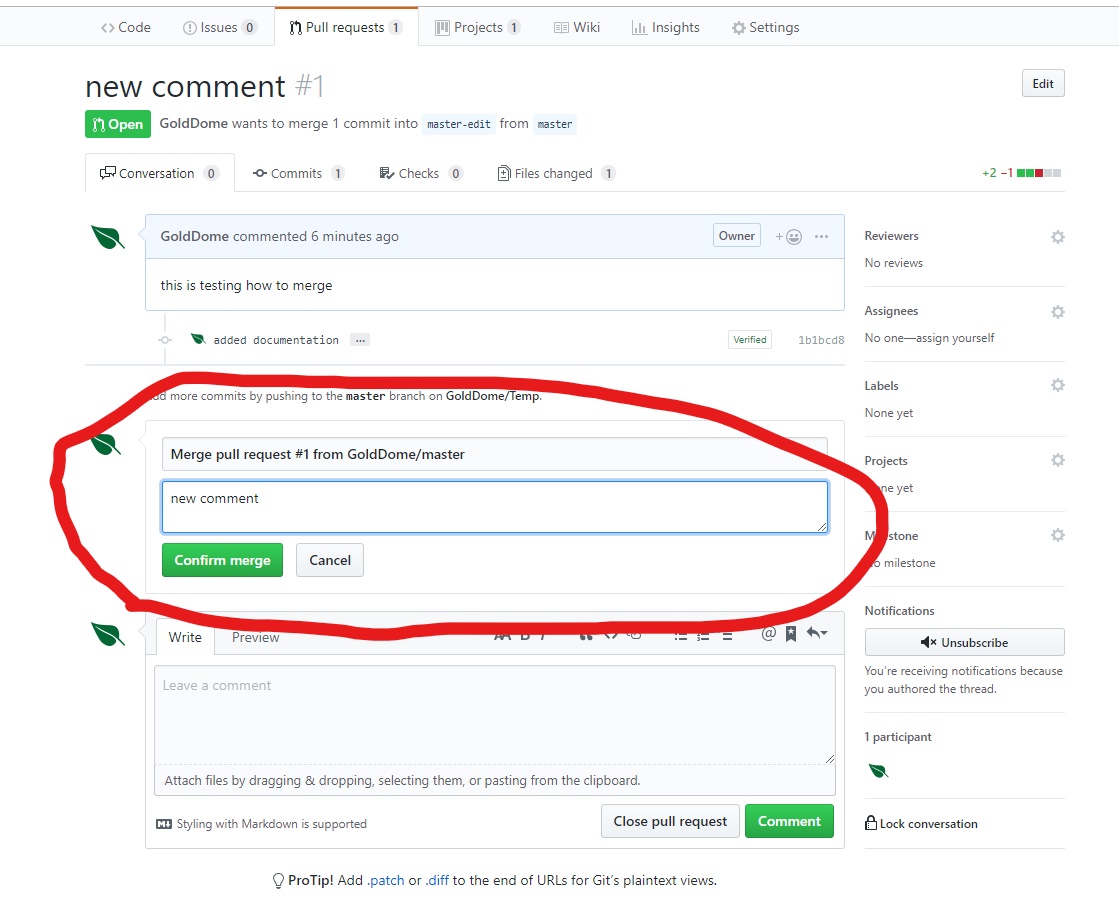
How to merge a pull request

-First you open your pull request tab and look for the pull request you want to merge

-Once you’ve found it you simply click “Merge Pull Request”



-After you merge the pull requests you are required to add a comment to describe the nature of why your merging the two branches.



Git Flow and other useful information

Now that we’ve gone over all the different aspects of GitHub, let’s review the flow. The Git Flow is all the steps you take to implement the different aspect.

-First you create a branch. Remember that branches are copies of their masters and they are used for edits. Branches allow for many people on a team to make parallel efforts towards a project. They’re also great for keep different members up to date with changes made to a project.

-Next you commit changes. To make changes to files in a repository, you make commits. Commits are saved changes to files in the repository. It’s important to document you commits so that other contributors and easily understand your changes. This is also important because commits are save in the project’s history, so they should be well documented.

-Next you open pull request, this is a request to merge two branches. These are ways to alert other members of a programming team that the changes are being finalized. They will have to be approved by the project owner before the branches can be merged.

-Before a pull request is approved, members of the team often meet and talk about the code that they are changing. This is always important to both to make sure that everyone on a team is on the same page but also to make sure that the code is being validated by every member of a team. If you are working on a project on GitHub individually, then you won’t need to meet a discuss pull requests, just review them on your own.

-Once you have discuss the code then you can merge two branches. Once you merge two branches then any changes made to another branch become permanent. It’s always important to comment what the nature of the change is so that it’s easier documented for other contributors to understand the changes.

-Finally, you implement your code from GitHub. You can download the files or copy and paste the code into your development software. From then the code is implemented and ready for use.