Standards and Practices for Git and Project Management

Andrew Sudano

August 18, 2016

* This should not be used as an introduction to Git and GitHub. Short hand notation is used without a full explanation of some of the functionality.

1 Branches

When you're working on this project, you're going to have a bunch of different features or ideas in progress at any given time – some of which are ready to go, and others which are not. Branching exists to help you manage this work flow.

When you create a branch in your project, you're creating an environment where you can try out new ideas and work on features. Changes you make on a branch don't affect the master branch, so you're free to experiment and commit changes while safe in the knowledge that your branch won't be merged until it's ready to be reviewed by someone you're collaborating with.

To create a branch enter the command listed below into your command terminal.Make sure to use the naming convention "issue-(IssueNumber)-(Title of Issue)"

```
git branch "issue-#(IssueNumber)-(Title of Issue)"
```

To then switch to that branch created enter the command below.

```
git checkout "issue-#(IssueNumber)-(Title of Issue)"
```

You have now switched to a new branch of the project and can begin working on implementing your new feature

2 Committing

Committing what you are working on to your local branch is one of the most important tasks for developer. Every time you finish a new component of your feature you should commit that change. In doing so you are providing your self a fail safe, in case anything goes wrong and you need to revert back to an older commit.

```
git commit -am "a detailed description of what you are committing"
```

* The above command is short hand combining both the add and commit message into one. This may not always be the way you want to commit code

3 Pull Requests

Pull Requests are commonly used by teams and organizations collaborating using the Shared Repository Model, where everyone shares a single repository and topic branches are used to develop features and isolate changes. Many open source projects on GitHub, and software companies use pull requests to manage changes from contributors as they are useful in providing a way to notify project maintainers about changes one has made and in initiating code review and general discussion about a set of changes before being merged into the main branch.

The code review allows for other developers to look at your code and make comments about its functionality and provide ideas for ways to improve it.

Although it may not seem that important, the Pull requests process is one of the most important tasks a software developer can do as it allows for a platform for developers to catch bugs before they are merged into the master branch and it is to late.

When you believe that you are done working on your code it's time to push your local branch to the master repository. You can, and it is advised, that you push your branch to the master repository periodically so that the list of branches can stay updated.

Before you push your branch you should pull the latest master branch to your local branch. This will not erase any of the code you were working on. It will just bring your branch in line with remote master branch.

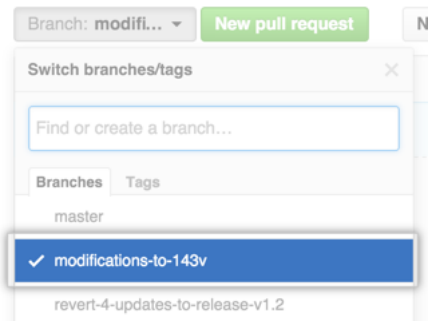
```
git pull upstream "branchname"
```

Now you push your branch to the remote repository

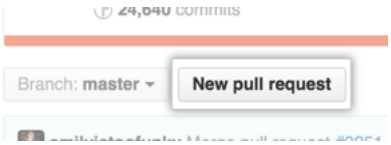
```
git push -u origin "Your Branch Name"
```

Now that you pushed your local branch to the GitHub repository it will contain your most recent commit.

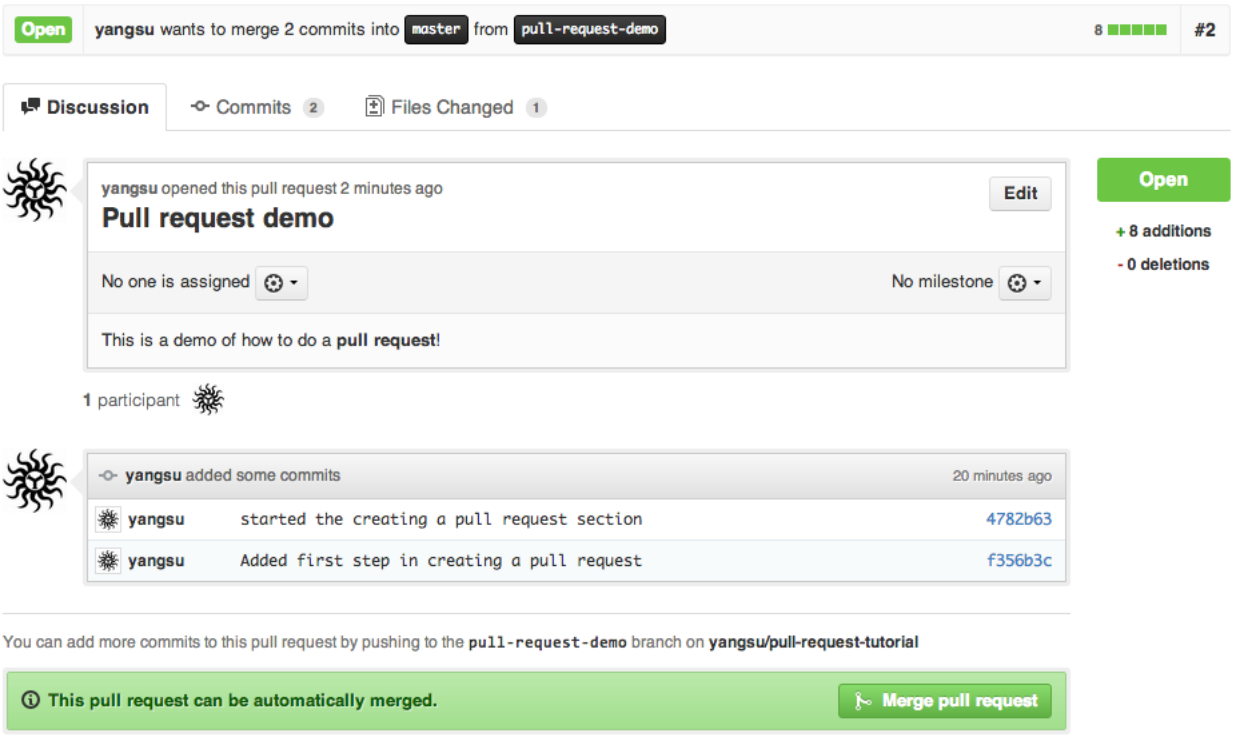
Now, if you go to the GitHub website you should be able to select your branch from the drop down list.



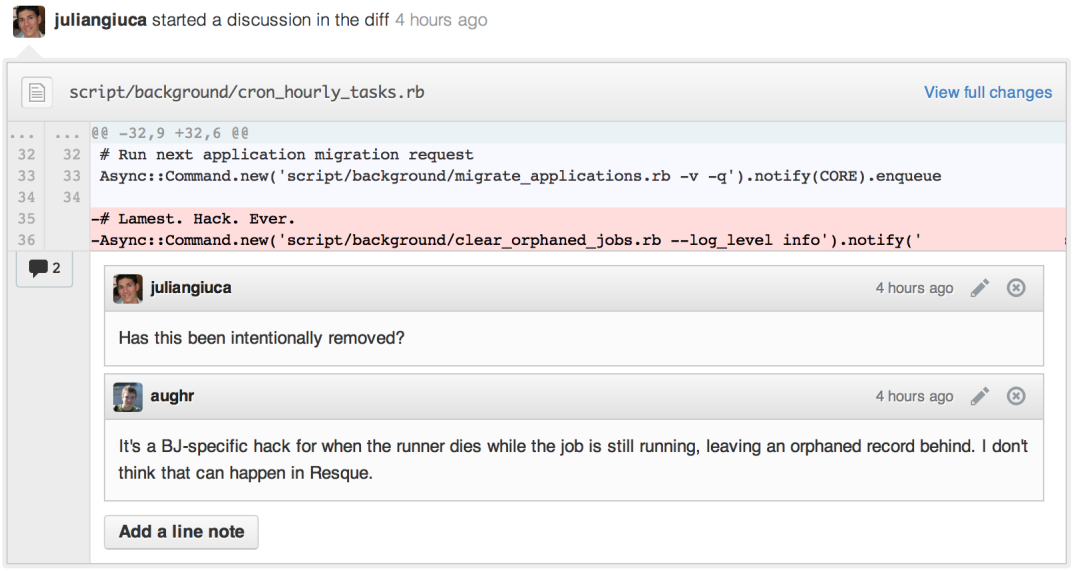
Once you select your branch you need to click "New pull request"



The following picture shows the screen that should come up ouce you initialize the pull request.



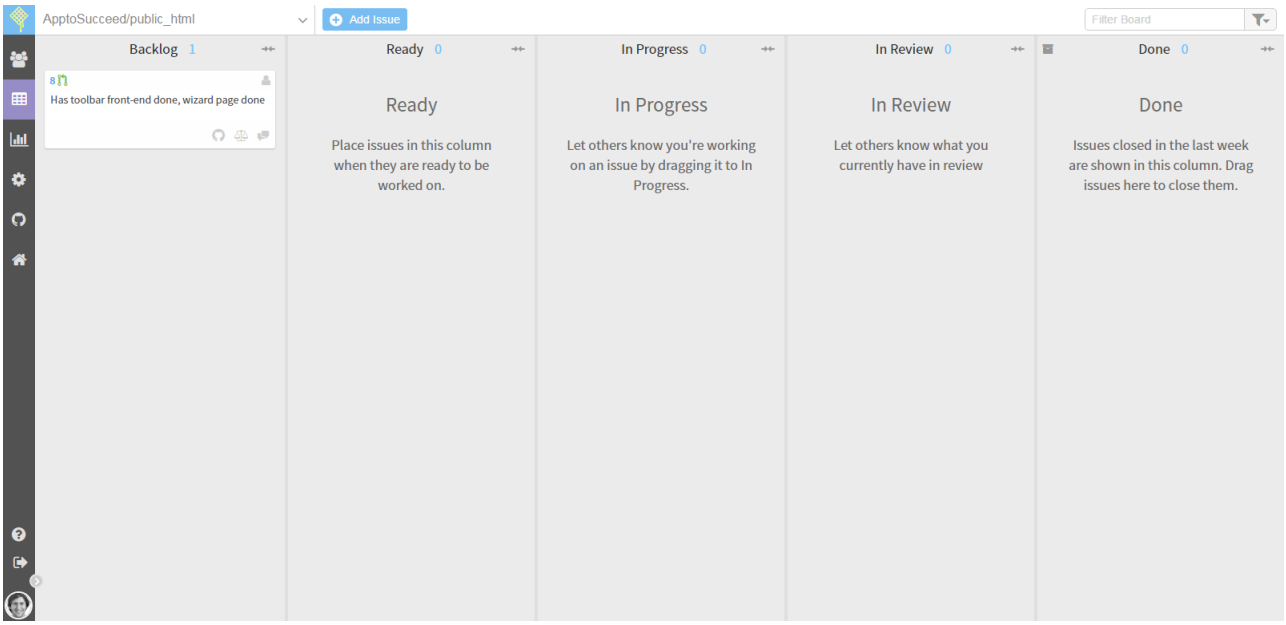
The following picture shows how someone would make comments on someone else's code.



For a branch to be merged into master it will require two reviewers to approve the pull request.

4 Project Management using waffle.io

Waffle.io gives development teams the ability to easily manage different development tasks and where those tasks are in the development cycle. The current URL for waffle.io is: https://waffle.io/ApptoSucceed/ApptoSucceedWebsite.This URL might change as GitHub is rearranged



Each of the columns have different meanings representing different stages in the development cycle. **Make sure to constantly be updating the board during the developmental cycle.**

4.1 Backlog

Features in this column are currently not ready to be worked on as they are being blocked by another issue. Once this issue can be worked on, it needs to be moved to the "Ready" column.

4.2 Ready

When you are ready to begin working on a task drag issues from the ready column to here. Features located here are ready to be worked on by any developer. Developers can pick and chose the issues they work on with nothing being explicitly assigned.

4.3 In Progress

Anything listed here is currently being worked on by a developer. No developer should ever have more than one feature in the "In Progress" column.

4.4 In Review

The Features listed here currently have pull request out in GitHub and are waiting for the necessary two approvals in order to be merged into master.

4.5 Done

Anything listed here has recently been completed and merged into master.

4.6 Adding Tasks

Everyone can and should be adding tasks to the board as they see necessary. Tasks added to the board are supposed to represent a new feature and or a bug to be worked on. Adding a new task should be done on GitHub. In the ApptoSucceed repository on the top tool bar to the right of code, there is a tab called "issues." Click on that and then go to "New Issue." Make a title that briefly explains the task and a description that explains what needs to be done and any files that apply. Once the issue is saved in GitHub waffle.io will be automatically updated.