

Git Branches

GITHUB PULL REQUEST, Branching, Merging & Team Workflow (11:20)

<https://www.youtube.com/watch?v=oFYyTZwMyAg>

Overview

This document describes why we branch, how to branch, and the commands we use to branch.

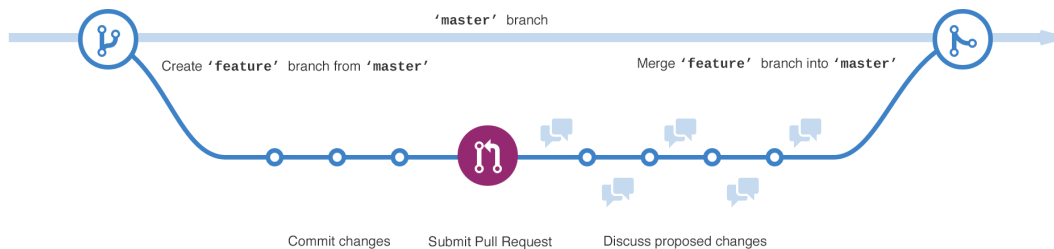
Branching

Why we branch

The [Git basics](#) documentation describes how to make changes to the github repository. Developers make changes to local copies of the repository, called branches. Branches allow developers to make changes to a local version of the repository, get those changes reviewed, and then submit the changes to the master branch. The reviewer attempts to catch any mistakes in the commit.

How to branch

Consider the repository to be the master branch, or `master`. Any changes made to newly created branches do not affect the `master`. When you create a branch, you create a copy of the repository.



<https://guides.github.com/activities/hello-world/branching.png>

Developers make their changes to these newly created branches, usually for a specific feature. Generally each JIRA story gets its own branch.

Before the developer pushes their branch to the master branch, they submit a `pull request`. A pull request requires another person to sign off for approval.

Commands to branch

Command	Function
<code>git branch</code>	list branches available
<code>git branch <name of branch></code>	create a new branch
<code>git checkout <name of branch></code>	swap between branches
<code>git checkout -b <name of branch></code>	create a new branch, then immediately swap to it
<code>git fetch</code>	If someone made a change to the <code>master</code> branch, download all of those changes to my local branch
<code>git merge</code>	*** finish this off but I think it is merge from master to my own branch

~~git pull~~

Do **not** use `git pull`, use `git fetch` **then** `git merge` instead. For more information see this website's <http://longair.net/blog/2009/04/16/git-fetch-and-merge/> article.

Procedure for pull request

Your team's assignment is to create documentation about how to use branches in git. You would use the following commands to execute this task.

1. Run `git checkout master` to ensure you are on the master branch
2. Run `git checkout -b BranchDocs` to create a new branch, and immediately switch to that branch
3. Create the document about branches, and save it in the repository's folder
 - a. For example, if you ran `git clone` in your local `~/Documents` folder, then repo-titled sub folder will be in the `~/Documents` folder
 - b. You may need to move the file to the sub folder, for example

```
mv ~/Documents/BranchBasics.pdf  
~/Documents/gitrepos/TestRepo/BranchBasics.pdf
```

4. Run `git checkout master`
 - a. Run `git fetch` then `git merge` to ensure you are up to date
 - b. Run `git checkout BranchDocs` then `git merge master` to pull any changes from master into your `BranchDocs` branch
5. Run `git add <filename>`, for example `git add BranchBasics.pdf`
6. Run `git commit`
7. Run `git push`