GitHub Commands

Initialize git:

$ git init

– Clone a repository:

$ git clone

– Make changes to a repository:

$ git add

$ git commit -m “message”

$ git push

See changes made:

$ git status

– Create a new branch:

$ git branch <new branch name>

– List all branches associated with the git repo:

$ git branch -a

– Move between branches:

$ git checkout <branch name>

– Create and checkout a new branch:

$ git checkout -b <new branch name>

– Merge a branch:

git merge <branch to merge> -m "message"

– Delete a branch (e.g., after it’s been merged):

$ git branch -d <branch to delete>

– See history of commits after adding commits to a new branch:

$ git log --graph

– Update a list of available branches at a remote repository:

git fetch <remote name>

– Store changes for later use to help with merge conflicts:

$ git stash

As a shorthand, you can fetch and then automatically merge the same-named branch from the remote using:

$ git pull <remote-name> <branch-name-to-merge-in>

Error:

Auto-merging <filename>

CONFLICT (content): Merge conflict in <filename>

Automatic merge failed; fix conflicts and then commit the result.

Solution:

What happened here? Well, you made changes to the same file (<filename> in this case) on both the base and the feature branches, and now Git doesn't know which set of changes to retain. In order to resolve this, you need to manually fix the relevant file. To do this, first open the relevant file. After opening the file, you will see conflict markers -- <<<<<<< HEAD, =======, and >>>>>>> branch-name. These markers indicate which portions of the file have conflicting changes. The markers and the conflicting changes are represented like this:

...

...

everything is fine here

<<<<<<< HEAD

...

...

this is the portion from the destination branch

=======

...

...

this is the portion from the source branch

>>>>>>> branch-name

All the lines between <<<<<<< HEAD and ======= are the conflicting lines from the base branch and all lines between ======= and >>>>>>> branch-name are the conflicting lines from the <branch-name> branch. Here you can discard the set of conflicting lines of your choice and then delete the conflict markers (<<<<<<< HEAD, =======, and >>>>>>> branch-name). After you do this, add and commit your changes so you can merge the branches without any issues.

Note: If there are multiple conflicts in the same file, each conflict is surrounded by conflict markers and it is your responsibility to resolve each conflict individually and remove all the conflict markers before you attempt to merge again.

>> git fetch upstream

>> git merge upstream/main

>> git status

>> git add .

>> git commit -m “message”

>> git status

>> git fetch upstream

>> git merge upstream/main

>> git pull origin main