

Git Workflow

Ideal Git Workflow:

1. Pull the most recent code from our Github Master Repo to your Local Master Branch with *git pull*
2. **Create a branch** for the code you're about to write (*git checkout -b [branch name]*)
3. Code + commit often (be sure to commit with *git push origin [branch name]*).
4. Once you're finished with your feature, checkout your Local Master branch and *git pull* the most recent code from the Github Master Repo.
5. *git checkout* your feature branch again and run *git merge master*. This will pull the most recent code from your Local Master branch to your Local Feature Repo. **Nobody should be working on the same files at the same time. This should avoid merge conflicts.**
6. Once you're finished testing your feature/branch with the most recent code on Master, do a final commit and create a **pull request**.
7. Head to github.com and find our repository online. You should be prompted to create a pull request for the GitHub repo branch you've been working on.
8. Create the pull request and send a message to everyone saying you need someone to look at the PR. Once we are happy with it, we merge the pull request.

"Oh shit, I've been coding on my local master branch"

Here's what to do.

1. Run *git stash*. This will take all your uncommitted stages (staged and unstaged) and save them.
2. Save your local master branch. Run *git status* and it should be a clean working directory.
3. Create a new branch for the feature you've been building on Master. *git checkout -b [feature-name]*
4. Run *git stash pop*. The changes you made on master are now in this branch. Save, commit, work as you please.