

Rafael Silva – [rafael.silva@fareoffice.com](mailto:rafael.silva@fareoffice.com)

Date:2017-11-28

Version: 1.0

Document: 024 – Git useful operations – Pt I - 171128

## Useful git flows of operations

1. Flow 1: Edit the message of the last commit both on your local repository and your remote repository.

Steps:

- Switch to your master branch (in case you have more than one branch).
- Edit the commit message.
- Push the changes to your remote repository.

Commands and actions in sequence:

- `git commit --amend -m "NEW/EDITED COMMIT MESSAGE"` (To edit the last commit message on your local repository)
- `git push --force` (To push the changes and update the remote repository).

2. Flow 2: Delete the last commit from the remote repository.

Steps:

- Reset the current HEAD to a specific state, locally.
- Delete the last commit from the remote repository.

Commands and actions in sequence:

- `git reset --hard HEAD~1` (To reset the HEAD of your local repo. In this case, we did reset it to the last-but-one commit!)
- `git push -f origin HEAD:master` (To delete the last commit of your remote repository)

3. Flow 3: Create a new branch, work on it and merge the changes to master (without merge conflicts. Not tracking remotely the branches).

Steps:

- Create a new branch.
- Switch to the new branch, e.g. “feature-branch”.
- Make changes on the files and, then, make a commit from your branch.
- Switch to the master branch.
- Merge the two branches.
- Push the merged branches to your remote repository.
- Visualize the tree with the branches on git bash.

Commands and actions in sequence:

- `git branch feature-branch.` (to create a new branch)
- `git branch` (to see the branches and which one is selected)
- `git checkout feature` (to switch to the feature branch)
- `git branch` (to see the branches and which is selected)

\* work on the feature branch by e.g. adding files, editing existing files, etc.

- `git add .` (to add your changes to your local HEAD)
- `git commit -m “YOUR COMMIT MESSAGE”` (to make a commit)
- `git checkout master` (to switch back to the master branch. Since we want to merge everything to the master branch, we need to be currently there!)
- `git merge feature-branch.` (to merge the changes that were made in the feature-branch into the master branch)
- `git log --graph --oneline --all` (To visualize the branches and flows on git bash)

NOTE: The changes of your feature branch are now merged with the existing content of the master branch. In order to continue working on your feature branch, you need again to switch to your feature branch.