

Split or merge repos

Merge repos into one

subtree add

Adding remote repos as subtrees results in merges in history, which cannot be rebased to get flat history. Rebasing causes to subtree appear in main repo root directory instead of subdirectory. Better use next method with moving files manually.

```
git remote add my-subtree /path/to/subtree.git
git subtree add --prefix=vendor/ my-subtree master
# with squashed commits # git subtree add --squash --prefix=vendor/ my-subtree master
# cleanup # git rebase my-subtree/master
```

Merge another repo (RRR) into directory of repo (LLL)

Idea is to fetch remote repo RRR into local repo LLL Then modify file location of repo BBB to be in desired path and merge it into repo LLL.

```
##### WAY 2
# Clone RRR repo temporary to change all file paths to new repo.
cd tmp
git clone http://URI_to_RRR_repo
git checkout main
# now move files from remote repo RRR into correct path:
mkdir app_RRR # create new location path (which will be merged to
destination RRR repo)
git checkout -b branch_change_file_location
git mv -k * app_RRR/src
git mv src app_RRR/src
git mv .gitignore .gitlab-ci.yml .gitmodules app_RRR/src

git commit -m "location of app_RRR adapted to local repo"

# optionally reword all commits:
git rebase --root -x 'git commit --amend'

cd LLL_repo
git remote add RRR /tmp/cloned_RRR_repo
git fetch RRR
git merge RRR/branch_change_file_location --allow-unrelated-histories
##### WAY 2 END
```

```

##### WAY 1
git remote add RRR http://URI_to_RRR_repo
git fetch RRR
git checkout -b branch_change_file_location RRR/main

# now move files from remote repo RRR into correct path:
mkdir app_RRR
git mv src app_RRR/src
git commit -m "location of app_RRR adapted to local repo"

# switch to LLL repo
git checkout master
# merge previously created branch "LLL/branch_change_file_location" into
"LLL/master"
git merge branch_change_file_location --allow-unrelated-histories
git commit
##### WAY 1 END

# cleanup
git remote rm RRR
git branch -d branch_change_file_location

# optionally - rebase:
# rebase can be problematic. Conflict will be on commonly used paths
(readme, Makefile, src, etc)
# between LLL and RRR repo
git rebase
# push merged master
git push

```

Split dirs into separate repos

git filter tools

https://github.com/slobobaby/git_filter

Detach(move) subdirectory into separate Git repository

<http://stackoverflow.com/questions/359424/detachmove-subdirectory-into-separate-git-repository>

```

# Split dir form repo
pushd <big-repo>
git subtree split -P <name-of-folder> -b <name-of-new-branch>
popd

```

```
# Create new repo
mkdir <new-repo>
pushd <new-repo>

git init
git pull </path/to/big-repo> <name-of-new-branch>

# Link the new repo to Github or wherever

git remote add origin <git@github.com:my-user/new-repo.git>
git push origin -u master

#Cleanup, if desired

popd # get out of <new-repo>
pushd <big-repo>

git rm -rf <name-of-folder>
```

Split one file from repo

```
# reduce repo to one folder
git filter-branch --prune-empty --subdirectory-filter etc -- --all

# reduce repo to some given file/files
git filter-branch -f --prune-empty --index-filter 'git rm --cached --ignore-unmatch $(git ls-files | grep -v "tmarc.xml|check-pazpar2.sh")'

# clean
git reflog expire --expire=now --all && git gc --prune=now --aggressive
```

From:

<https://niziak.spox.org/wiki/> - niziak.spox.org

Permanent link:

https://niziak.spox.org/wiki/git:split_merge_repos

Last update: 2024/01/17 11:48

