Forking a sub directory of a repository on GitHub and making it part of my own repo

Asked 8 years, 9 months ago Modified 4 years, 4 months ago Viewed 19k times

41

Apologies, I am very new to Git and GitHub, I've read through a few things but I'm not sure if what I'm trying to do is *entirely* possible.

Basically I want to fork the Confluence Skin used on XBMC and modify various elements located here:

https://github.com/xbmc/xbmc/tree/master/addons/skin.confluence

However I don't want to fork the entire XBMC repository, so a simple fork action won't do.

Here are my general requirements:

- I would like to take the contents of the skin.confluence folder and put it into a repository on my own GitHub account
- I need to be able to keep it linked within the original XBMC repo to receive upstream commits, as my modifications will generally be theme based, rather than functionality.

Thanks to the response posted by tbekolay, I have been able to do a subtree split to take only the skin.confluence part of repo and essentially create a branch, however I am unsure on how to keep it linked with the original XBMC repo while being pushed to my own repo with my modifications.

git github repository git-subtree xbmc

Share Follow

edited Feb 6, 2017 at 17:10
approxiblue
6,912 16 50 59

asked Jul 4, 2014 at 15:06
James White
666 1 10 18

I don't think that is possible. It is possible with other version control systems, but git only allows cloning entire repositories. You could try convincing the xmbc team to split the skin out into a separate repository. – Thayne Jul 4, 2014 at 15:12

@Thayne So subversion might be better for what I required in this case? Or just use Git but manually pull in changes from upstream? – James White Jul 4, 2014 at 15:17

Yes. Assuming an SVN repository is maintained. – Thayne Jul 4, 2014 at 15:20

1 I've rewritten my answer with more details about how you should set up the repositories, and how to receive upstream commits. Whether it's worth the hassle is up to you ;) The commands take a lot of time to run, but I think it's done somewhat cleanly. – tbekolay Jul 10, 2014 at 11:50

@tbekolay Thank you for taking the time to update your previous answer since my original posting. I will go through everything now and see! – James White Jul 10, 2014 at 12:42

1 Answer

Sorted by: Highest score (default) ⬍

83

+50

This is mostly possible with `git`, though it is not a typical use case, so there may be some rough edges as you do this (especially if you are new to `git`).

The tool that we'll use for this job is `git subtree`.

Setting up a repository

Start by cloning the whole XBMC repository.

```
git clone https://github.com/xbmc/xbmc.git
cd xbmc
```

We start on the `master` branch by default. We want to make our own `master` branch, so let's rename `master` to `upstream-master`.

```
git branch -m upstream-master
```

Now use `git subtree split` to only include the part that you want. We'll make the split off part a new branch called `upstream-skin`.

```
git subtree split --prefix=addons/skin.confluence -b upstream-skin
git checkout upstream-skin
```

This gives you a new `upstream-skin` branch that only contains the contents of `addons/skin.confluence`, and with a filtered history that contains only the commits that modified files in `addons/skin.confluence`.

Now, let's set up our remotes. Since you cloned `xbmc/xbmc.git`, the `origin` remote will point there. Let's rename that to `upstream`.

```
git remote rename origin upstream
```

Make a repository on Github to contain your modifications to `addons/skin.confluence`. As an example, I'll use tbekolay/xbmc-skin, but replace this with your own repo. Add this repo as a remote, and push your `upstream-skin` branch to it.

```
git remote add origin https://github.com/tbekolay/xbmc-skin.git
git fetch origin
git push -u origin upstream-skin
```

Finally, we'll make a new branch called `master` that will contain your changes.

```
git checkout -b master
git push -u origin master
```

You now have a "fork" of the `addons/skin.confluence` subdirectory.

Making changes to your repositories

When you're dealing with your own local and remote repositories, you can use normal `git` commands. Make sure to do this on the `master` branch (or some other branch, if you'd like) and not the `upstream-skin` branch, which should only ever contain commits from the upstream project.

```
git checkout master
echo "My XBMC Skin" > README
git add README
git commit -m "Added README"
git push
```

Receiving upstream commits

When you're dealing with the upstream repository, you will have to use a mix of `git` and `git subtree` commands. To get new filtered commits, we need to do it in three stages.

In the first stage, we'll update `upstream-master` to the current version of the XBMC repository.

```
git checkout upstream-master
git pull
```

This should pull down new commits, if there are any.

Next, we will update `upstream-skin` with the new filtered version of the commits. Since `git subtree` ensures that commit hashes will be the same, this should be a clean process. Note that you want to run these commands while **still on the `upstream-master` branch**.

```
git subtree split --prefix=addons/skin.confluence \
  --onto upstream-skin -b upstream-skin
```

With `upstream-skin` now updated, you can update your `master` branch as you see fit (either by merging or rebasing).

```
git checkout master
git rebase upstream-skin
```

Note that the XBMC repository is gigantic, and the `git subtree` commands will take quite a bit of time to filter through all that history -- and since you're regenerating the split subtree each time you interact with the remote repository, it's quite an expensive operation. I'm not sure if this can be sped up.

This blog post goes into some more detail on the commands above. Also see the `git-subtree` docs for even more detail.

Share Follow

edited Nov 22, 2018 at 16:57

answered Jul 4, 2014 at 15:19
tbekolay
16.7k 3 39 37

Thanks for this, I kept seeing subtree and submodules mentioned, I'll check this out and see what I can do. – James White Jul 4, 2014 at 15:25

2 Definitely stay away from submodules for a number of reasons, the most salient of which being that you would have to make changes to the XMBC repository in order to use it. – tbekolay Jul 4, 2014 at 15:34

Thanks for the heads up! – James White Jul 4, 2014 at 15:36

@tbekolay, thx, this is really useful. Question: if I want to make a collection of sub directories from a few different upstream, is that possible? I see the fist command is `git clone` , which makes forking from several different upstreams difficult. Please answer here -> stackoverflow.com/questions/45201567. thx. – xpt Jul 19, 2017 at 21:45

3 The blog post has sadly disappeared. Can be found on the wayback machine here: web.archive.org/web/20131123125622/http://blog.charlescy.com/... – Damian Dixon Nov 14, 2018 at 8:05