

# Github 上 fork 项目后与原项目保持同步

标签: [iOSgithubandriod](#) [创建 github 分支与合并](#)

2016-05-03 15:26 2737 人阅读 [评论\(0\)](#) [收藏](#) [举报](#)

分类:

github 相关操作

版权声明: 本文为博主原创文章, 未经博主允许不得转载。

[目录\(?\)](#)[+](#)

其实网上这类似的教程很多, 但是由于每个描述不同, 大家接受的方式也不同

有时候我们会发现, 在几个月前 **fork** 来的代码, 在原作者更新后, 我们享受不到更新带来的便利。

一般有两种做法, 第一种做法可谓简单粗暴, 但却不是最好的做法, 说明如下:

1.将自己仓库中原来 **fork** 的项目删掉 (**delete deletethis repository**), 然后在重新到原作者仓库中(**fork**)一份最新代码 (**不推荐方式**)

这似乎达到了我们的最终目的, 虽然有时候我们做事是为了达到某种目的而不看过程, 但不尽然, 有时候复杂的的过程

往往会伴随着你意想不到的收获, 为什么有人被称之为神 (O(∩\_∩)O 哈哈~, 开个玩笑), 当然这得看每个人的生活方

式了, 总仁者见仁智者见智, 这里我不多做赘述。

2. 方法是在本地建立两个库的中介，把两个远程库都 clone 到本地，然后拉取原项目更新到本地，合并更新，最后 push 到你的 github 就完成。（推荐方式）

（1）准备一个本地目录，并克隆自己 fork 的项目到本地。

```
1$ git clone https://github.com/jp1017/FastAndroid.git
```

（2）然后 cd 进入仓库（这里我直接进入项目根目录下），执行命令：git remote -v

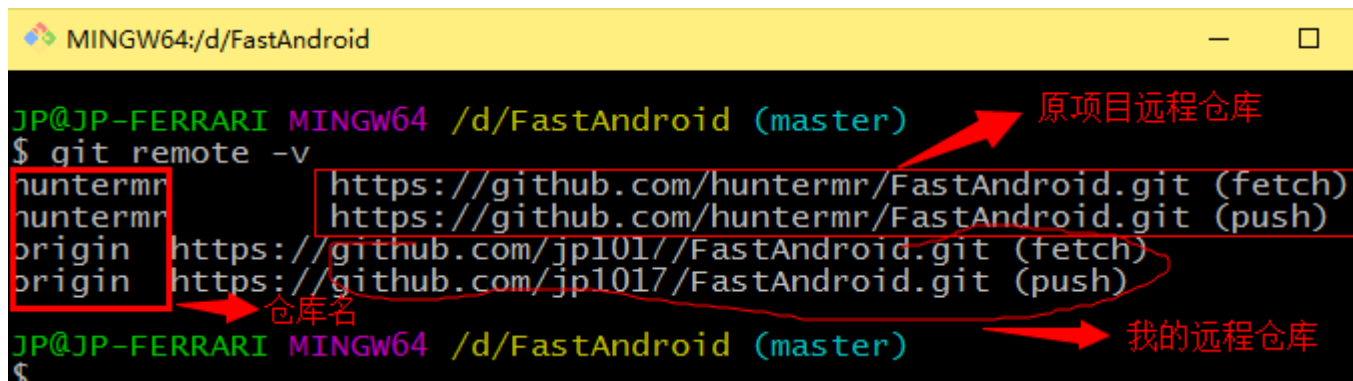
```
1$ git remote -v
```

```
JP@JP-FERRARI MINGW64 /e/FastAndroid (master)
$ git remote -v
origin  https://github.com/jp1017/FastAndroid.git (fetch)
origin  https://github.com/jp1017/FastAndroid.git (push)
```

（3）我们可以看到，只有我们自己的远程仓库，下面 clone 原项目到该仓库

```
1$ git remote add hunter https://github.com/huntermr/FastAndroid.git
```

```
2$ git remote -v
```



The screenshot shows a terminal window with the following output:

```
JP@JP-FERRARI MINGW64 /d/FastAndroid (master)
$ git remote -v
huntermr https://github.com/huntermr/FastAndroid.git (fetch)
huntermr https://github.com/huntermr/FastAndroid.git (push)
origin   https://github.com/jp1017/FastAndroid.git (fetch)
origin   https://github.com/jp1017/FastAndroid.git (push)
```

Annotations in the image:

- A red box highlights the `huntermr` remote name, with a red arrow pointing to it and the label "仓库名" (Repository Name).
- A red box highlights the URLs for the `huntermr` remote, with a red arrow pointing to it and the label "原项目远程仓库" (Original project remote repository).
- A red box highlights the URLs for the `origin` remote, with a red arrow pointing to it and the label "我的远程仓库" (My remote repository).

（4）这个 hunter 名字随便取哦，这次有两个远程分支咯，我们继续 fetch

然后把原项目更新的内容 fetch 到本地：git fetch hunter

```
1$ git fetch hunter
```

（5）查看下分支：

1\$ git branch -av

```
$ git branch -av
* master          fcb4259 Merge remote-tracking branch 'hunter/master'
remotes/hunter/master 597d144 添加网络访问okhttp支持 把这个合并
remotes/origin/HEAD -> origin/master
remotes/origin/master fcb4259 Merge remote-tracking branch 'hunter/master'
/master'
```

(6) 一个本地分支 master，三个远程分支，画红线的就是要合并的 merge

1\$ git checkout master

2\$ git merge hunter/master

```
$ git checkout master
Already on 'master'
Your branch is up-to-date with 'origin/master'.

JP@JP-FERRARI MINGW64 /e/FastAndroid (master)
$ git merge hunter/master
Already up-to-date.
```

---

push

1\$ git push -u origin master

(7) merge 后本地的仓库已经是最新的咯，然后 push 到你的 github 就同步完成。

## 1.可能使用到的一些 git 命令

检查本地仓库 git 状态    `git status`

添加仓库修改            `git add .` (添加所有修改) / `git add ~/file`(添加具体文件)

确认提交修改            `git commit` 这里会出现一个面板，里边主要用来说明本次修改的相关信息

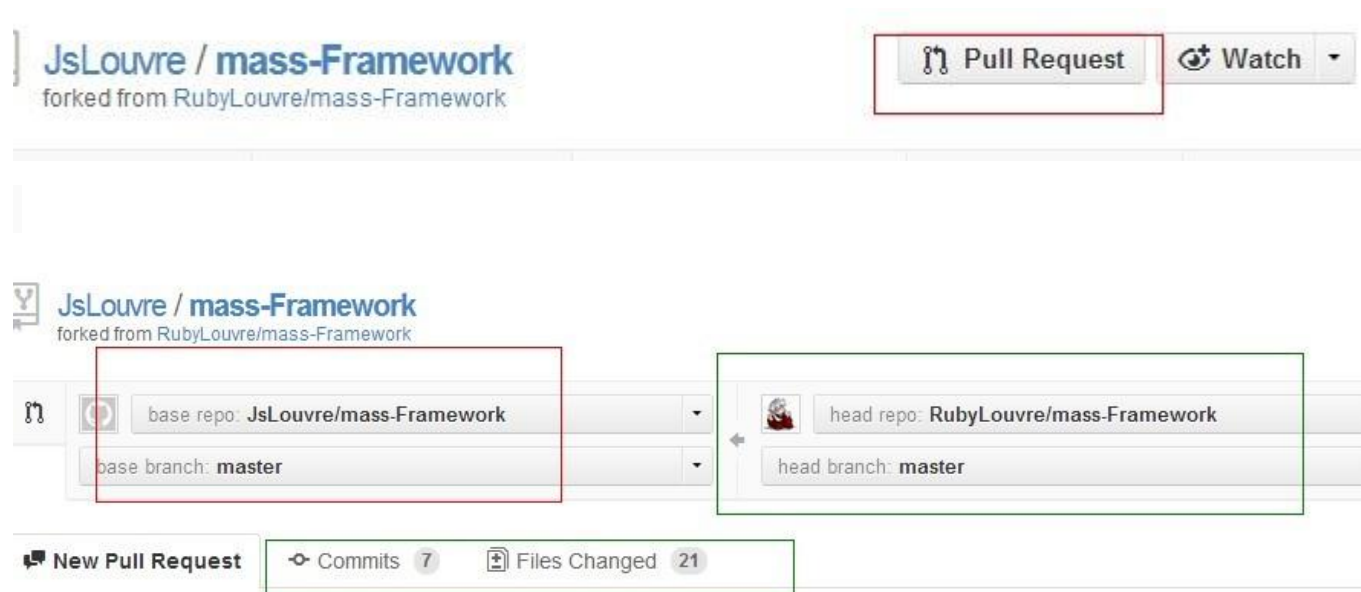
检查更新                `git checkout`

将对本地仓库的修改提交到 github 仓库 `git push -u origin master`

# 线上办法：

如何让自己的项目与原作者的项目保持同步！

我经常看到许多人，只会 fork 一次，提交过一次修改就不知怎么办了！因此原框架作者是非常勤奋的，一天会提交 N 次，一个星期后许多文件都改动过了，而那些代码贡献者不可能一个个跟着修改。 因此我们还是用到上方的 pull request 按钮。



这次是把自己的项目放到左边，原框架作者放到右边，在选择过程中，你会发现原框架作者有许多贡献者的。这里希望大家一定要浏览 **Commits** 与 **Files Changed** 进行学习！这也是 github 最大的价值所在！把握别人对代码的改进，最能提高我们编码水平。这里会涉及大量的编码技巧！

 base repo: JsLouvre/mass-Framework

head repo: RubyLouvre/m

base branch: master

head branch: master

## New Pull Request

Commits 7

Files Changed 21

合并原作者的更新

Write


Preview

Comments are parsed with GitHub Flavored Markdown



如题


补上必要说明，然后点击下方 send pull request 按钮。


很快你就看到评论区最下方有个鲜绿色高亮的按钮，继续点就是。


No one is assigned 


如题


2 participants  





 RubyLouvre added some commits


 RubyLouvre Merge pull request #39 from JsLouvre/master ...


 RubyLouvre 添加教程!

 RubyLouvre Merge branch 'master' of github.com:RubyLouvre/mass-Framework


 RubyLouvre 更新readme

 RubyLouvre update 图片!

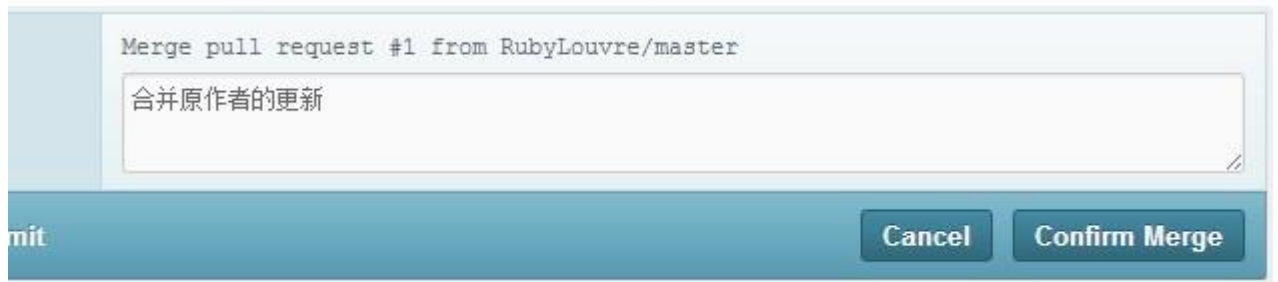
 RubyLouvre update readme!

 RubyLouvre 将注释移到内部

This pull request can be automatically merged.

 Merge

继续让你确认，没问题就继续点！



这样就同步成功！！！！！！

有时候我们在 github 上看到一些比较好的项目，我们都会 fork 一下它，然后在本地进行操作，但是 fork 之后，项目是不会跟源项目保持同步的，需要我们自己进行一些操作让其同步。我们以最近 Nutz 成员正在开发的 QA 系统为例，假如我们 clone 自己 fork 过来的项目的保存路径为：~/ngqa\_gevinhgy

具体操作如下：

- 1、打开 Git Bash
- 2、进入~/ngqa\_gevinhgy
- 3、为了让项目与源项目同步，我们首先要将源项目 pull 到本地

```
1 git remote add ngqa https://github.com/howe/ngqa.git #添加一个别名，指向 ngqa 项目的位置
```

```
2 git pull ngqa master #pull ngqa 下面的 master 分支
```

```
Administrator@NC-GEVIN ~/ngqa_gevinhgy (master)
$ git remote add ngqa https://github.com/howe/ngqa.git

Administrator@NC-GEVIN ~/ngqa_gevinhgy (master)
$ git pull ngqa master
From https://github.com/howe/ngqa
* branch          master      -> FETCH_HEAD
Already up-to-date.

Administrator@NC-GEVIN ~/ngqa_gevinhgy (master)
$
```

ps：由于我刚刚执行过，因此显示的是全部都是最新版本。

- 4、将 pull 下来的代码 push 到自己的项目上面

```
l git push
```

问题描述： 当我们 在 **github** 上 **fork** 出一个项目后，如果原有的项目更新了，怎样保持我们 **fork** 出来的项目和原有项目保持同步呢并提交我们的代码更新呢？即怎样保持 **fork** 出的项目和上游项目保持更新，怎样创建 **pull request**？关键步骤是使用 **git** 的 **rebase** 命令。

步骤：

1. 在 **Fork** 的代码库中添加上游代码库的 **remote** 源，该操作只需操作一次即可。

如：其中 **# upstream** 表示上游代码库名， 可以任意。

```
git remote
```

```
add upstream https://github.scm.corp.ebay.com/montage/frontend-ui-  
workspace
```

2. 将本地的修改提交 **commit**

3. 在每次 **Pull Request** 前做如下操作，即可实现和上游版本库的同步。

- 3.1 : `git remote update upstream`

- 3.2 : `git rebase upstream/{branch name}`

需要注意的是在操作 3.2 之前，一定要将 **checkout** 到 **{branch name}** 所指定的 **branch**,

如: `git checkout develop`

4. **Push** 代码到 **Github**

```
git push
```