Git 由浅入深之远端主机（git remote）

我们知道无论是分布式版本控制系统还是集中式版本控制系统，如果要实现多人协作，都需要一个远程服务器，具体针对某一项目来说，就是一个远程仓库。

无论使用什么版本控制工具，对于每一个成员而言，无外乎就是共享数据（push or pull），而这些协作都需要通过一个处于远端主机上的远端仓库完成。本篇主要介绍 Git 如何与远端主机进行操作，主要包括：添加和移除远端主机，添加远程仓库，管理远程分支等。

添加远端主机（git remote add / git clone）

我们首先要了解如何显示地添加一个远端主机，在 Git 中有两种方式添加远端主机：可以分为显式或隐式。

在未添加远端主机之前，查看主机信息为空：

```
jiangangxiong@jiangangxiong MINGW64 /d/work/ser (master)
$ git remote

jiangangxiong@jiangangxiong MINGW64 /d/work/ser (master)
$ git remote -v

jiangangxiong@jiangangxiong MINGW64 /d/work/ser (master)
```

隐式添加（git clone）

隐式添加远端主机的方式就是使用 git clone <远端地址>指令，在克隆远端仓库的同时，会自动添加该远端主机到当前目录，并且默认主机名为 origin:

```
jiangangxiong@jiangangxiong MINGW64 /d/work/ser (master)
$ git clone https://github.com/codingplayboy/spa
Cloning into 'spa'...
remote: Counting objects: 20, done.
remote: Total 20 (delta 0), reused 0 (delta 0), pack-reused 20
Unpacking objects: 100% (20/20), done.
Checking connectivity... done.
```

查看信息：

```
jiangangxiong@jiangangxiong MINGW64 /d/work/ser (master)
$ git remote
spa

jiangangxiong@jiangangxiong MINGW64 /d/work/ser (master)
$ git remote -v
spa     https://github.com/codingplayboy/spa (fetch)
spa     https://github.com/codingplayboy/spa (push)
```

显示添加（git remote add）

很多时候我们也需要显示添加远端主机，这需要使用 git remote add <主机别名> <远端地址> 指令显式添加一个远端主机：

```
jiangangxiong@jiangangxiong MINGW64 /d/work/ser/spa (master)
$ git remote add react https://github.com/codingplayboy/reactjs

jiangangxiong@jiangangxiong MINGW64 /d/work/ser/spa (master)
```

指令中，主机别名参数为自定义指定，远端地址即远端服务器上的访问地址。

再查看其主机信息，则会有如下显示：

```
jiangangxiong@jiangangxiong MINGW64 /d/work/ser/spa (master)
$ git remote
origin
react

jiangangxiong@jiangangxiong MINGW64 /d/work/ser/spa (master)
$ git remote -v
origin  https://github.com/codingplayboy/spa (fetch)
origin  https://github.com/codingplayboy/spa (push)
react   https://github.com/codingplayboy/reactjs (fetch)
react   https://github.com/codingplayboy/reactjs (push)
```

另一点我们需要清楚的是，可以为某一目录指定任意数量远端主机。

查看远端主机信息（git remote）

我们可以使用 git remote 指令查看当前仓库指定的所有远端主机的简要信息：别名列表，当然还有添加-v 选项，查看详细信息列表：


```
jiangangxiong@jiangangxiong MINGW64 /d/work/ser/spa (master)
$ git remote
origin
react

jiangangxiong@jiangangxiong MINGW64 /d/work/ser/spa (master)
$ git remote -v
origin  https://github.com/codingplayboy/spa (fetch)
origin  https://github.com/codingplayboy/spa (push)
react   https://github.com/codingplayboy/reactjs (fetch)
react   https://github.com/codingplayboy/reactjs (push)
```


查看特定主机信息（git remote show）

我们也可以查看某一特定远端主机的详细信息，使用 git remote show <主机别名>如：

```
jiangangxiong@jiangangxiong MINGW64 /d/work/ser/spa (master)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/codingplayboy/spa
  Push  URL: https://github.com/codingplayboy/spa
  HEAD branch: master
  Remote branch:
    master tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local ref configured for 'git push':
    master pushes to master (up to date)
```

如上，我们可以看到详细信息，远端地址，远程所有分支，
及在各本地分支上使用 git pull 或 git push 指令操作时对应的远程分支。

数据共享（git fetch & git pull & git push）

添加远程主机后，我们可以与主机共享代码或文件。

git fetch

使用 git fetch <主机名> [远程分支，可选]:[新建本地分支，可选]指令，即会将该主机地址对应的远程仓库中所有数据（包括所有分支）拉取到本地，前面提到的隐式添加远端主机 git clone 方式已包含这一过程：

```
$ git fetch react
warning: no common commits
remote: Counting objects: 110, done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 11Receiving objects:  89% (98/110), 1.60 (delta 37 MiB | 4.00 KiB/
2), reused 110 (delta 32), pack-reused 0
Receiving objects: 100% (110/110), 1.67 MiB | 4.00 KiB/s, done.
Resolving deltas: 100% (32/32), done.
From https://github.com/codingplayboy/reactjs
 * [new branch]      master     -> react/master
```

如上，我们已经将远端 react 仓库的所有分支，下载到本地的对应分支（此处只有 master 分支，其下载到本地的 react/master 分支），我们可以将这些本地分支合并到其他分支或在这些本地分支的基础上检出（check out）新分支。
当指定远程分支名时，如下：

```
jiangangxiong@jiangangxiong MINGW64 /d/work/ser/spa/react (master)
$ git fetch react master:temp
From https://github.com/codingplayboy/reactjs
 * [new branch]      master     -> temp

jiangangxiong@jiangangxiong MINGW64 /d/work/ser/spa/react (master)
```

将远端 origin 仓库的 master 分支下载到本地并新建一个本地分支 temp。

git pull

除了使用 git fetch 指令拉取远程数据，还可以使用 git pull <主机名> <远程分支>指令拉取远程特定分支的所有数据：

```
$ git pull react master
From https://github.com/codingplayboy/reactjs
 * branch            master     -> FETCH_HEAD
Auto-merging README.md
CONFLICT (add/add): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
```

其与 git fetch 的差别有两点：

- gi pull 需要指定特定远程分支参数
- git pull 指令会自动拉取数据并将其合并至当前分支，而 git fetch 只是拉取所有数据及分支，不影响本地数据，我们需要手动合并。

git push

git fetch 及 git pull 对应的另一个指令则是 git push <主机名> <本地分支>，使用该指令可以向远端推送分支数据：

```
jiangangxiong@jiangangxiong MINGW64 /d/work/ser/spa (xjg-temp)
$ git push origin master
Everything up-to-date

jiangangxiong@jiangangxiong MINGW64 /d/work/ser/spa (xjg-temp)
$
```

重命名远端主机别名（git remote rename）

Git 也支持我们重命名之前添加过的远端主机别名：

git remoet rename <旧主机别名> <新别名>

```
jiangangxiong@jiangangxiong MINGW64 /d/work/ser/spa (master)
$ git remote
origin
react
test

jiangangxiong@jiangangxiong MINGW64 /d/work/ser/spa (master)
$ git remote rename test pre

jiangangxiong@jiangangxiong MINGW64 /d/work/ser/spa (master)
$ git remote
origin
pre
react
```

删除远端主机别名（git remote remove|rm）

使用 git remote remove|rm <主机别名>指令删除远端主机：