

git remote 命令管理一组跟踪的存储库。

要参与任何一个 Git 项目的协作，必须要了解该如何管理远程仓库。远程仓库是指托管在网络上的项目仓库，可能会有好多个，其中有些你只能读，另外有些可以写。同他人协作开发某个项目时，需要管理这些远程仓库，以便推送或拉取数据，分享各自的工作进展。管理远程仓库的工作，包括添加远程库，移除废弃的远程库，管理各式远程库分支，定义是否跟踪这些分支等等。

使用语法

```
git remote [-v | --verbose]
git remote add [-t <branch>] [-m <master>] [-f] [--[no-]tags] [--mirror=<fetch|push>] <name> <url>
git remote rename <old> <new>
git remote remove <name>
git remote set-head <name> (-a | --auto | -d | --delete | <branch>)
git remote set-branches [--add] <name> <branch>...
git remote get-url [--push] [--all] <name>
git remote set-url [--push] <name> <newurl> [<oldurl>]
git remote set-url --add [--push] <name> <newurl>
git remote set-url --delete [--push] <name> <url>
git remote [-v | --verbose] show [-n] <name>...
git remote prune [-n | --dry-run] <name>...
git remote [-v | --verbose] update [-p | --prune] [(<group> | <remote>)...]
```

描述

git remote 命令管理一组跟踪的存储库。

示例

以下是一些示例 -

1.查看当前的远程库

要查看当前配置有哪些远程仓库,可以用 **git remote** 命令,它会列出每个远程库的简短名字.在克隆完某个项目后,至少可以看到一个名为 **origin** 的远程库, git 默认使用这个名字来标识你所克隆的原始仓库:

```
$ git clone http://git.oschina.net/yiibai/sample.git
$ cd sample
```

(1)**git remote** 不带参数，列出已经存在的远程分支

```
$ git remote origin
```

2) `git remote -v | --verbose` 列出详细信息, 在每一个名字后面列出其远程 url 此时, `-v` 选项(译注:此为 `--verbose` 的简写,取首字母),显示对应的克隆地址:

```
$ git remote -v
origin http://git.oschina.net/yiibai/sample.git (fetch)
origin http://git.oschina.net/yiibai/sample.git (push)

Administrator@MY-PC /D/worksp/sample (master)
$ git remote --verbose
origin http://git.oschina.net/yiibai/sample.git (fetch)
origin http://git.oschina.net/yiibai/sample.git (push)
```

2.添加一个新的远程, 抓取, 并从它检出一个分支 -

```
$ git remote
origin
$ git branch -r
  origin/HEAD -> origin/master
  origin/master
$ git remote add staging git://git.kernel.org/.../gregkh/staging.git
$ git remote
origin
staging
$ git fetch staging
...
From git://git.kernel.org/pub/scm/linux/kernel/git/gregkh/staging
 * [new branch]      master      -> staging/master
 * [new branch]      staging-linus -> staging/staging-linus
 * [new branch]      staging-next -> staging/staging-next
$ git branch -r
  origin/HEAD -> origin/master
  origin/master
  staging/master
  staging/staging-linus
  staging/staging-next
$ git checkout -b staging staging/master
...
```

3.添加远程仓库

要添加一个新的远程仓库,可以指定一个简单的名字,以便将来引用,运行 `git remote add [shortname] [url]`:

```
$ git remote
  origin
$ git remote add pb http://git.oschina.net/yiibai/sample.git
$ git remote -v origin http://git.oschina.net/yiibai/sample.git
  pb http://git.oschina.net/yiibai/sample2.git 现在可以用字符串 pb 指代对应的仓库地址了.
比如说,要抓取所有 Paul 有的,但本地仓库没有的信息,可以运行 git fetch pb:
```

```
$ git fetch pb
  remote: Counting objects: 58, done.
  remote: Compressing objects: 100% (41/41), done.
  remote: Total 44 (delta 24), reused 1 (delta 0)
  Unpacking objects: 100% (44/44), done.
  From http://git.oschina.net/yiibai/sample2.git
  * [new branch] master -> pb/master
  * [new branch] ticgit -> pb/ticgit
```

4.模仿 `git clone`, 但只跟踪选定的分支

```
$ mkdir project.git
$ cd project.git
$ git init
$ git remote add -f -t master -m master origin git://example.com/git.git/
$ git merge origin
```