

当你从远程仓库克隆时，实际上 Git 自动把本地的 `master` 分支和远程的 `master` 分支对应起来了，并且，远程仓库的默认名称是 `origin`。

要查看远程库的信息，用 `git remote`：

```
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (master)
$ git remote
origin
```

```
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (master)
$ git remote -v
origin git@github.com:ZebinGao/bluetooth.git (fetch)
origin git@github.com:ZebinGao/bluetooth.git (push)
```

上面显示了可以抓取和推送的 `origin` 的地址。如果没有推送权限，就看不到 `push` 的地址。

推送分支，就是把该分支上的所有本地提交推送到远程库。推送时，要指定本地分支，这样，Git 就会把该分支推送到远程库对应的远程分支上

```
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (master)
$ git push origin master
Everything up-to-date
```

如果要推送其他分支，比如 `dev`，就改成

```
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (master)
$ git push origin dev
Total 0 (delta 0), reused 0 (delta 0)
To github.com:ZebinGao/bluetooth.git
2ee2e28..5ff1a19 dev -> dev
```

现在，模拟一个你的小伙伴，可以在另一台电脑（注意要把 SSH Key 添加到 GitHub）或者同一台电脑的另一个目录下克隆：

```
admin@ZEBIN-GAO MINGW64 /d/BlueTryTest
$ git clone git@github.com:ZebinGao/bluetooth.git
Cloning into 'bluetooth'...
remote: Enumerating objects: 83, done.
```

```
remote: Counting objects: 100% (76/76), done.
remote: Compressing objects: 100% (55/55), done.
remote: Total 83 (delta 29), reused 67 (delta 20), pack-reused 7 (from 1)
Receiving objects: 100% (83/83), 1.73 MiB | 106.00 KiB/s, done.
Resolving deltas: 100% (30/30), done.
```

当你的小伙伴从远程库 clone 时，默认情况下，你的小伙伴只能看到本地的 master 分支。不信可以用 git branch 命令看看：

```
admin@ZEBIN-GAO MINGW64 /d/BlueTryTest
$ git branch
fatal: not a git repository (or any of the parent directories): .git

admin@ZEBIN-GAO MINGW64 /d/BlueTryTest
$ ls
bluetooth/

admin@ZEBIN-GAO MINGW64 /d/BlueTryTest
$ cd bluetooth

admin@ZEBIN-GAO MINGW64 /d/BlueTryTest/bluetooth (master)
$ git branch
* master
```

现在，你的小伙伴要在 dev 分支上开发，就必须创建远程 origin 的 dev 分支到本地，于是他用这个命令创建本地 dev 分支：

```
admin@ZEBIN-GAO MINGW64 /d/BlueTryTest/bluetooth (master)
$ git checkout -b dev origin/dev
Switched to a new branch 'dev'
Branch 'dev' set up to track remote branch 'dev' from 'origin'.
```

现在，他就可以在 dev 上继续修改，然后，时不时地把 dev 分支 push 到远程：

```
admin@ZEBIN-GAO MINGW64 /d/BlueTryTest/bluetooth (dev)
$ git add .
admin@ZEBIN-GAO MINGW64 /d/BlueTryTest/bluetooth (dev)
$ git commit -m "add 001 chapter in dev branch"
```

```
[dev 359c2bb] add 001 chapter in dev branch
4 files changed, 0 insertions(+), 0 deletions(-)
admin@ZEBIN-GAO MINGW64 /d/BlueTryTest/bluetooth (dev)
$ git push origin dev
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 182.78 KiB | 763.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:ZebinGao/bluetooth.git
5ff1a19..359c2bb dev -> dev
```

现在，他就可以在 `dev` 上继续修改，然后，时不时地把 `dev` 分支 `push`

到远程：

```
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (dev)
$ git add .

admin@zebin MINGW64 /d/BlueTestTry/bluetooth (dev)
$ git commit -m "remove test.txt"
[dev 93ae214] remove test.txt
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 test.txt

admin@zebin MINGW64 /d/BlueTestTry/bluetooth (dev)
$ git push origin dev
To github.com:ZebinGao/bluetooth.git
! [rejected]        dev -> dev (fetch first)
error: failed to push some refs to 'git@github.com:ZebinGao/bluetooth.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

推送失败，因为你的小伙伴的最新提交和你试图推送的提交有冲突，

解决办法也很简单，Git 已经提示我们，先用 `git pull` 把最新的提交

从 `origin/dev` 抓下来，然后，在本地合并，解决冲突，再推送：

```
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (dev)
```

```
$ git pull
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 5 (delta 2), reused 5 (delta 2), pack-reused 0 (from 0)
Unpacking objects: 100% (5/5), 182.76 KiB | 331.00 KiB/s, done.
From github.com:ZebinGao/bluetooth
    5ff1a19..359c2bb dev      -> origin/dev
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.
```

```
git pull <remote> <branch>
```

If you wish to set tracking information for this branch you can do so with:

```
git branch --set-upstream-to=origin/<branch> dev
```

`git pull` 也失败了，原因是没有指定本地 `dev` 分支与远程 `origin/dev`

分支的连接，根据提示，设置 `dev` 和 `origin/dev` 的连接：

```
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (dev)
$ git branch --set-upstream-to=origin/dev dev
Branch 'dev' set up to track remote branch 'dev' from 'origin'.
```

## 再 pull

```
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (dev)
$ git pull
Removing 重新开始未完成的工作/~WRL0003.tmp
Removing 重新开始未完成的工作/~$5-分支管理.docx
```

```
Merge branch 'dev' of github.com:ZebinGao/bluetooth into dev
# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
```

```
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (dev)
$ git commit -m "conflict fixed"
On branch dev
Your branch is ahead of 'origin/dev' by 2 commits.
(use "git push" to publish your local commits)
```

nothing to commit, working tree clean

```
admin@zebin MINGW64 /d/BlueTestTry/bluetooth (dev)
$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
writing objects: 100% (4/4), 482 bytes | 482.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:ZebinGao/bluetooth.git
   359c2bb..eafec06 dev -> dev
```

因此，多人协作的工作模式通常是这样：

1. 首先，可以尝试用 `git push origin <branch-name>` 推送自己的修改；
2. 如果推送失败，则因为远程分支比你的本地更新，需要先用 `git pull` 试图合并；
3. 如果合并有冲突，则解决冲突，并在本地提交；
4. 没有冲突或者解决掉冲突后，再用 `git push origin <branch-name>` 推送就能成功！

如果 `git pull` 提示 `no tracking information`，则说明本地分支和远程分支的链接关系没有创建，用命令 `git branch --set-upstream-to <branch-name> origin/<branch-name>`。