



Tracking Remote Branches With Different Names

Learn how to track remote branches that have different names from local branches.

We'll cover the following



- Tracking a branch
- Reproducing the situation

Tracking a branch

More rarely, you may want to track a branch on the remote repository that has a different name. Or you may want to manually mark the local branch as tracking a specific remote one.

In these situations, you might see this kind of error when you push:

```
1 git push
```

Terminal 1



Terminal



Click to Connect...



As is often the case, a careful reading of the error will tell you what you need to know. It's just the jargon that can be difficult to follow!

In this case, the error is telling you that your branch is not tracking any remote branch, so it doesn't know what to push to.

Reproducing the situation

Type in these commands to reproduce this situation:

```
2  rm -rf git_origin git_clone
3  mkdir git_origin
4  cd git_origin
5  git init
6  echo 'first commit' > file1
7  git add file1
8  git commit -am file1
9  cd ..
10 git clone git_origin git_clone
11 cd git_clone
12 git checkout -b abranch
13 echo 'origin abranch commit' >> file1
14 git commit -am 'cloned abranch commit'
15 git push
```

Terminal 1

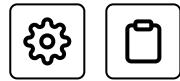


Terminal

^

Take a look at the branches you have locally when you try to push:

```
16 git branch -vv
```



While the master branch is tracking the origin/master branch (i.e., the master branch on the origin remote), the abranch branch is not tracking any remote branch.

At this point, you could run either:

```
17 git push --set-upstream origin abranch
```

Or

```
18 git push -u origin abranch
```

And that would set up the tracking for you while pushing.

Before that though, you're going to type:

```
19 git push origin abranch
```

Terminal 1



Terminal

^

Can you see the difference?

That successfully pushed the change to the remote branch, which was created as if it did not already exist. However, if you re-run the branch command again:

```
20 git branch -vv
```

It is still not tracking the origin's master branch. If you add the --set-upstream / -u flag on a push, the branch will track the remote's branch:

```
21 git push -u origin abranch  
22 git branch -vv
```

[← Back](#)[Next →](#)

The Branch Exists Only On The Remote

Introduction: Git Submodules

[Mark as Completed](#)[Report an Issue](#)[Ask a Question](#)

(https://discuss.educative.io/tag/tracking-remote-branches-with-different-names__pushing-code__learn-git-the-hard-way)