Andhra Pradesh State Skill Development Corporation

Skill AP
APSSDC

eSkill AP

Learn Anytime Anywhere

Andhra Pradesh State Skill Development Corporation

Source Code Management Using Git & GitHub

Git Remotes

Working with Remotes

To be able to collaborate on any Git project, you need to know how to manage your remote repositories. Remote repositories are versions of your project that are hosted on the Internet or network somewhere. You can have several of them, each of which generally is either read-only or read/write for you. Collaborating with others involves managing these remote repositories and pushing and pulling data to and from them when you need to share work. Managing remote repositories includes knowing how to add remote repositories, remove remotes that are no longer valid, manage various remote branches and define them as being tracked or not, and more. In this section, we'll cover some of these remote-management skills.

showing remotes

To see which remote servers you have configured, you can run the git remote command. It lists the short names of each remote handle you've specified. If you've cloned your repository, you should at least see origin — that is the default name Git gives to the server you cloned from:

```
git clone https://github.com/schacon/ticgit
Cloning into 'ticgit'...
remote: Reusing existing pack: 1857, done.
remote: Total 1857 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (1857/1857), 374.35 KiB | 268.00 KiB/s, done.
Resolving deltas: 100% (772/772), done.
Checking connectivity... done.
$ cd ticgit
$ git remote
origin
```

You can also specify -v, which shows you the URLs that Git has stored for the short name to be used when reading and writing to that remote:

```
git remote -v
originhttps://github.com/schacon/ticgit
(fetch)
originhttps://github.com/schacon/ticgit
(push)
```

If you have more than one remote, the command lists them all. For example, a repository with multiple remotes for working with several collaborators might look something like this.

```
git remote -v
bakkdoor https://github.com/bakkdoor/grit
(fetch) bakkdoor
https://github.com/bakkdoor/grit (push) cho45
          https://github.com/cho45/grit (fetch)
cho45     https://github.com/cho45/grit (push)
defunkt https://github.com/defunkt/grit (fetch)
defunkt https://github.com/defunkt/grit (push)
koke      git://github.com/koke/grit.git (fetch)
koke      git://github.com/koke/grit.git (push)
origin    git@github.com:mojombo/grit.git
(fetch) origin
          git@github.com:mojombo/grit.git
(push)
```

Adding a remote

To add a new remote to the repository

```
git remote add <shortname> <url>
```

Send the files from local to the remote repository

When you have your project at a point that you want to share, you have to push it upstream. The command for this is simple:

```
git push <remote> <branch>
```

For Example, my remote name is origin5 and the branch is master so the command would be like this

```
git push origin5 master
```

If there are already some files present in the repository and if you are sending the files then your push will rightly be rejected. So, first, you should have to synchronize the remote repository and local repository

git pull <remotename> <branch>

Renaming and RemovingRemotes

To rename any remote the command is

git remote rename <old remote name> <new remote name>

Remove the remote

If you want to remove a remote for some reason — you've moved the server or are no longer using a particular mirror, or perhaps a contributor isn't contributing anymore

```
git remote remove <remote name>
```