

Join the Stack Overflow Community

Stack Overflow is a community of 6.9 million programmers, just like you, helping each other. Join them; it only takes a minute:

[Sign up](#)

How to link a folder with an existing Heroku app



I have an existing Rails app on GitHub and deployed on Heroku. I'm trying to set up a new development machine and have cloned the project from my GitHub repository. However, I'm confused as to how to link this folder up to Heroku. Originally, I used the "heroku create" command, but obviously I don't want to do that since it will create another Heroku instance.

[git](#) [heroku](#)

edited Aug 31 '11 at 19:24



[Purplejacket](#)
450 3 22

asked Feb 26 '11 at 20:39



[Kevin Pang](#)
21.4k 34 110 160

5 Answers

[Heroku](#) links your projects based on the `heroku git remote` (and a few other options, see the update below). To add your Heroku remote as a remote in your current repository, use the following command:

```
git remote add heroku git@heroku.com:project.git
```

where `project` is the name of your Heroku project (the same as the `project.herokuapp.com` subdomain). Once you've done so, you can use the `heroku xxxx` commands (assuming you have the [Heroku Toolbelt](#) installed), and can push to Heroku as usual via `git push heroku master`. As a shortcut, if you're using the command line tool, you can type:

```
heroku git:remote -a project
```

where, again, `project` is the name of your Heroku project (thanks, [Colonel Panic](#)). You can name the Git remote anything you want by passing `-r remote_name`.

[Update]

As mentioned by Ben in the comments, the remote doesn't need to be named `heroku` for the gem commands to work. I checked [the source](#), and it appears it works like this:

1. If you specify an app name via the `--app` option (e.g. `heroku info --app myapp`), it will use that app.
2. If you specify a Git *remote* name via the `--remote` option (e.g. `heroku info --remote production`), it will use the app associated with that Git remote.
3. If you specify no option and you have `heroku.remote` set in your Git config file, it will use the app associated with that remote (for example, to set the default remote to "production" use `git config heroku.remote production` in your repository, and Heroku will run `git config heroku.remote` to read the value of this setting)
4. If you specify no option, the gem finds no configuration in your `.git/config` file, and the gem only finds one remote in your Git remotes that has "heroku.com" in the URL, it will use that remote.
5. If none of these works, it raises an error instructing you to pass `--app` to your command.

edited Jan 12 '13 at 16:21

answered Feb 26 '11 at 21:03



[Michelle Tilley](#)
110k 23 274 251

The link to "the source", above, doesn't work. Maybe try this page: devcenter.heroku.com/articles/using-the-cli -- though I think the doc there is not enough. -- [Purplejacket](#) Aug 31 '11 at 19:03

Thanks, Purplejacket, they renamed a module and the link was invalidated. I've updated the URL. -- [Michelle Tilley](#) Sep 1 '11 at 1:30

- 1 If you try to use the `heroku git:remote` method, check that you're using the heroku toolbelt cli, and that you don't have any heroku gem still installed on your system (seems it didn't work for me with the old gems). -- [PA. Buisson](#) Jan 11 '13 at 10:47
- 2 Also, for multiple heroku branches you can use `$ git config heroku.remote <apname>` to set default app so you can use terminal commands for that app. -- [Sheharyar](#) Jul 18 '13 at 16:27
- 1 if using http i.e. `heroku login` then the remote is of the form `https://git.heroku.com/project-name.git` -- [maxple](#) Aug 11 '16 at 0:33

The [Heroku CLI](#) has an easy shortcut for this. For an app named 'falling-wind-1624':

```
$ heroku git:remote -a falling-wind-1624
Git remote heroku added.
```

See <https://devcenter.heroku.com/articles/git#creating-a-heroku-remote>

edited Jun 24 '13 at 10:33

answered Aug 24 '12 at 19:52



[Colonel Panic](#)

59.5k 38 243 302

- 1 ! Resource not found -- [Green](#) Jul 28 '13 at 22:10

this still works. -- [Sgnl](#) Sep 21 '16 at 20:53

You can name the remote anything you want by passing `-r other_remote_name` -- [MasterScrat](#) Feb 4 at 14:35

This is still the correct way to do it even if you've connected your Heroku app directly to GitHub. Note it will not overwrite your "origin" remote; it creates a "heroku" remote instead. -- [Seth](#) Mar 8 at 1:32

Don't forget, if you are also on a machine where you haven't set up heroku before

```
heroku keys:add
```

Or you won't be able to push or pull to the repo.

answered Apr 29 '11 at 14:07



[Ghoti](#)

2,071 1 12 17

- 2 Oh yes! That helped me too. I had an app where we started basic, then bifurcated "heroku" into "staging" and "production". So I did `git remote add staging git@staging.xx:yy.git` and `git remote add production git@production.xx:yy.git` -- but that wasn't enough. When I tried to do `git push staging master` (or production) I'd get "Permission denied (publickey). fatal: The remote end hung up unexpectedly". What fixed that error was to do `heroku keys:add` as Ghoti mentions. -- [Purplejacket](#) Aug 31 '11 at 19:07

- 1 +1 thank you for this! -- [andy](#) Mar 21 '13 at 23:34

Thank you sir! -- [Br3x](#) Sep 20 '14 at 20:47

exactly what i needed -- [astroanu](#) May 10 '15 at 9:27

Use heroku's fork

1. Use the new "heroku fork" command! It will copy all the environment and you have to update the github repo after!

```
heroku fork -a sourceapp targetapp
```

2. Clone it local

```
git clone git@heroku.com:youamazingapp.git
```

3. Make a new repo on github and add it

```
git remote add origin https://github.com/yourname/your_repo.git
```

4. Push on github

```
git push origin master
```

answered Aug 11 '13 at 18:10



[msroot](#)

638 9 10

You should probable start ssh-agent and add your keys. Check this,

<http://wordgraphs.com/post/5000/Heroku--Permission-denied--publickey---fatal--Could-not-read-from-remote-repository->

It helped me.

answered Sep 15 '14 at 19:24



Stranger

3,599

4

39

75
