cobyism / **gh-pages-deploy.md**

Last active yesterday • Report abuse

☆ Star

<> **Code** Revisions 6 ☆ Stars 2,119 Forks 325

Deploy to `gh-pages` from a `dist` folder on the master branch. Useful for use with [yeoman] (http://yeoman.io).

gh-pages-deploy.md

Deploying a subfolder to GitHub Pages

Sometimes you want to have a subdirectory on the `master` branch be the root directory of a repository's `gh-pages` branch. This is useful for things like sites developed with Yeoman, or if you have a Jekyll site contained in the `master` branch alongside the rest of your code.

For the sake of this example, let's pretend the subfolder containing your site is named `dist`.

Step 1

Remove the `dist` directory from the project's `.gitignore` file (it's ignored by default by Yeoman).

Step 2

Make sure git knows about your subtree (the subfolder with your site).

```
git add dist && git commit -m "Initial dist subtree commit"
```

Step 3

Use subtree push to send it to the `gh-pages` branch on GitHub.

```
git subtree push --prefix dist origin gh-pages
```

Boom. If your folder isn't called `dist` , then you'll need to change that in each of the commands above.

If you do this on a regular basis, you could also create a script containing the following somewhere in your path:

```
#!/bin/sh
if [ -z "$1" ]
then
  echo "Which folder do you want to deploy to GitHub Pages?"
  exit 1
fi
git subtree push --prefix $1 origin gh-pages
```

Which lets you type commands like:

```
git gh-deploy path/to/your/site
```

Load earlier comments...

tyleryoungblood commented on Jan 29, 2021

> This has been working out for me;
>
> ```
> "scripts": {
>   ...
>   "deploy-demo": "git push origin :gh-pages && git subtree push --prefix dist
> origin gh-pages"
> }
> ```
>
> deletes the branch and re-pushes. Avoids all that remote branch being ahead fluff that others are mentioning.

@mikeyhogarth Awesome solution! Thank you! And I'd never seen that method of deleting a branch before.

However, is there a way to delete the remote branch, but only if it already exists? If you run `npm run deploy-demo` for the first time, it will error out because the gh-pages branch doesn't exist yet. It works perfectly for future deployments, just not the first one.

kutsan commented on Feb 17, 2021

Using https://www.npmjs.com/package/gh-pages was the way to go for me. Saves tons of headaches especially if you're working with others.

```
"scripts": {
  ...
  "deploy": "gh-pages --dist 'dist' --branch 'release'"
}
```

alex-drocks commented on Oct 17, 2021

> Using https://www.npmjs.com/package/gh-pages was the way to go for me. Saves tons of headaches especially if you're working with others.
>
> ```
> "scripts": {
>   ...
>   "deploy": "gh-pages --dist 'dist' --branch 'release'"
> }
> ```

This cleared my remote project and left only the dist files lol

kutsan commented on Oct 17, 2021

@alex-drocks This is how it was suppose to do. Some build systems produce file names with hashes in them for caching. In order to prevent duplication, files are on specified branch needs to be deleted before new ones take over.

My suggestion is, use a separate branch for dist files that is generated by your build system and configure GitHub Pages to use root (/) of that branch. Have a good day!

TimMTech commented on Feb 4, 2022

Hands down one of the worst git deployment step by steps I have ever encountered.

cobyism commented on Feb 4, 2022

> Hands down one of the worst git deployment step by steps I have ever encountered.

@TimMTech Considering this gist of mine is ~8 years old, I'll be the first admit there are many ways it is far from perfect. For what it's worth though, you'll probably get a better response from people when communicating online if you can provide constructive suggestions for improvement alongside any criticism/feedback you decide to share. 💟

goelshivam1210 commented on Feb 6, 2022

I want to deploy my website using Jekyll-scholar on Github pages. Since Github pages don't deploy it on their own, I will have to use another way of deploying a pre-built website. Can you please suggest some ways how to do that? I used rakefile, and when I force the built _site subdirectory to be the root of the project, it deletes everything, and therefore, nothing can be pushed.
I am using this command `git filter-branch --subdirectory-filter _site/ -f"`
I am using macOS 11.6 and using Jekyll-scholar. I am extremely new to this, so please pardon my naivety.

Azer5C74 commented on Apr 4, 2022

Thanks for this helpful contribution, I recently wanted to follow the same steps hoping to deploy a sub folder containing index.tsx file as a main file for a react typescript project, but it didn't work out for me even I succeed to push the subfolder into the remote gh-pages branch.

ichenpipi commented on Apr 7, 2022

Thanks, this solution work well for my subfolder(`build/web-mobile`).

I did it like:

```
git subtree push --prefix build/web-mobile github gh-pages
```

przemollo commented on Apr 25, 2022

@cobyism Thanks for this thread. At the beggining it didn't help me, but i spend couple of hours thinking and searching what is wrong.

I was getting errors in console (net::ERR_ABORTED 404). Then i realised that I have wrong path in my index.html to .js and .css files. They havn't got "./" at the beggining. I changed this and TA-DAH! It works!
I'm so happy now! Thank You!

fabriziomartini commented on Jun 16, 2022

`"git push origin :gh-pages && git subtree push --prefix dist origin gh-pages"` - this command worked perfectly

VenexCon commented on Jul 11, 2022

Does anyone know if this tutorial works if you are not dual-booting a linux OS? Say if i have created this project in a windows directory would this still work as intended? Having some issues figuring this out, as I cannot seem to find a good solution on how to publish this to gh-pages, nor if I was to publish it to another host i.e. Netlify or similar.

hosja83 commented on Jul 16, 2022

If you already have a "gh-pages" branch, use the 1st command below. If you don't have a "gh-pages" branch, initialize it by using the 2nd command below. Make it easier to run by creating scripts like these in your package.json file:

```
"scripts": {
      "gh-deploy": "git push origin :gh-pages && git subtree push --prefix dist origin
gh-pages"
      "gh-deploy-init": "git push origin && git subtree push --prefix dist origin gh-
pages",
}
```

In terminal:

```
npm run gh-deploy
npm run gh-deploy-init
```

We aren't done just yet. Now go to your repository in GitHub. Go to Settings. Go to Pages. Under the Source section you will see a dropdown list of branches. Select the "gh-pages" branch and select the root as your folder. Then click Save.

Give GitHub some time to rebuild and deploy your repository's site. Boom you have your site up and running.

Hopefully this works for you all. Shoutout to **@TheOdinProject**.

riivanov commented on Aug 14, 2022 • edited ▾

I tried this approach, but I think `git-worktree` , and deployment from a seperate branch, is a cleaner alternative, since I won't have commits in my `main` branch, intermingled with commits from re-deployment, which I find more succinct, and I won't have to delete the remote branch every time, which is unnecessary.

`git-worktree` mounts your sub-directory, `dist` , in this example, to a separate branch, `gh-pages` .

Here's how:

```
git branch --track gh-pages origin/gh-pages          # Create new gh-pages branch; Add tra
git checkout --orphan gh-pages                       # Initialize gh-pages without main's
git reset --hard                                     # Remove all history
git commit --allow-empty -m "Init"                  # First commit without any files
git checkout main                                   # Go back to main (or master) branch
git worktree add dist gh-pages                      # Mount dist and bind it to the gh-pa
```

`dist` is an `npm` build script which looks like:

```
"scripts": {
  ...
  "dist": "ng build --configuration production && echo 'gitdir: /home/<user>/<repo>/.git/
  ...
}
```

All it does is re-creates the `git-worktree` reference, because the `.git` file in `dist` was removed by `ng build` by default.
This reference is needed by `git` to link `dist` to the index.

And the workflow goes something like this:

```
npm run dist                                        # Build website with new changes; Rem
cd dist                                             # Move to gh-pages branch by switchin
git add .                                           # Add all generated files to staging
git commit -m "v0.0.3"                              # Update the version history of the g
git push                                            # Push changes to gh-branch
```

If you run `git status` it will reply `On branch  gh-pages` .
And `git log` will show one commit `"Init"` .

But when you `cd ..` and run `git status` again, the response will be `On branch main` .
And `git log` will show all of your original commits to `main` .

So what's happened here is quite interesting. The folder `dist` now has a separate branch, with it's own, unrelated history to `main` ,
and all you have to do to switch is `cd dist` to access that branch (`gh-pages`).

This is unlike `git checkout dist` , which would append the `dist` directory, with the auto generated build files to your working tree, intermingling your `main` and deployment histories, which is inconvenient.

Here your `src` files will be untouched, along with their own history in `main` , or `cd ..` , and only the files needed for deployment, will be on this branch, which is really convenient, because it keeps the `src` history seperate from the deployment history.

Now you'd deploy **not** from a folder, but from a branch, which holds the latest compiled version of your site in GitHub pages.

Of course there's probably an improvement that could be done here as well.
For example make `npm run dist` do all of this, but my personal preference is to do these steps manually.

Read more about this method here.

This is the how, for the suggestion by **@kutsan**.

> **@alex-drocks** This is how it was suppose to do. Some build systems produce file names with hashes in them for caching. In order to prevent duplication, files are on specified branch needs to be deleted before new ones take over.
>
> My suggestion is, use a separate branch for dist files that is generated by your build system and configure GitHub Pages to use root (/) of that branch. Have a good day!

After posting I realized **@ChrisBAshton** had already documented this approach. The only difference being the `echo` command in the `npm build dist` script.

But I'd agree, that if you're working on a team, it's probably better to use a tool like gh-pages, to enforce standards in your project.

I hope my explanation is somewhat of a contribution as well, and not just a re-statement of the mentioned methods above.

antariksh17 commented on Aug 22, 2022

head is genuinely spinning trying to understand this

antariksh17 commented on Aug 22, 2022 • edited ▾

i have a github repo - project
the subfolder , which is a create react app is named application

now how should i host my github pages from project/application?

riivanov commented on Aug 23, 2022

@antariksh17 if that's where the statically produced webpack(or other) bundles are output, then, you'll have a separate branch, in the above example called `gh-pages` , in your GitHub repo, which will only contain the static bundle files built from the source files of your `project/application` .

But I'm guessing, knowing no react, that `project/application` is not the same as `dist` , which is where the bundles will be output. So probably something like `project/application/dist` , is the folder, when mounted to your local git index, where your `ng build` react equivalent command would put your webpack bundles intended for distribution.

So if you did this correctly, then, `cd ~/project/application/dist` will display a separate version history, **as if** you'd done a `git init` in `project/application/dist`, and had a `.git` folder there, in addition to the parent folder. You can think about that `gh-pages` branch as a separate repo, and this is what makes this approach so convenient, because each `git push` from `project/application/dist` will only update `gh-pages`, with the latest compiled bundles, and none of the source files from `project`.

So you'd want to deploy from a branch in GitHub, `gh-pages` in the example above, using `/root`.

bomszx commented on Sep 14, 2022

> ```
> "scripts": {
>       "gh-deploy": "git push origin :gh-pages && git subtree push --prefix dist origin gh-pages"
>       "gh-deploy-init": "git push origin && git subtree push --prefix dist origin gh-pages",
> }
> ```
>
> In terminal:
>
> ```
> npm run gh-deploy
> npm run gh-deploy-init
> ```

this one worked for me, thanks mate!

SonyDennisAdams commented on Sep 21, 2022

> I want to deploy my website using Jekyll-scholar on Github pages. Since Github pages don't deploy it on their own, I will have to use another way of deploying a pre-built website. Can you please suggest some ways how to do that? I used rakefile, and when I force the built _site subdirectory to be the root of the project, it deletes everything, and therefore, nothing can be pushed. I am using this command `git filter-branch --subdirectory-filter _site/ -f"` I am using macOS 11.6 and using Jekyll-scholar. I am extremely new to this, so please pardon my naivety.

It's the leading underscore -- see: https://github.com/orgs/community/discussions/23640

SookiezBoly commented on Sep 26, 2022 • edited

> If you already have a "gh-pages" branch, use the 1st command below. If you don't have a "gh-pages" branch, initialize it by using the 2nd command below. Make it easier to run by creating scripts like these in your package.json file:

> ```
> "scripts": {
>       "gh-deploy": "git push origin :gh-pages && git subtree push --prefix dist
> origin gh-pages"
>       "gh-deploy-init": "git push origin && git subtree push --prefix dist origin
> gh-pages",
> }
> ```
>
> In terminal:
>
> ```
> npm run gh-deploy
> npm run gh-deploy-init
> ```
>
> We aren't done just yet. Now go to your repository in GitHub. Go to Settings. Go to Pages. Under the Source section you will see a dropdown list of branches. Select the "gh-pages" branch and select the root as your folder. Then click Save.
>
> Give GitHub some time to rebuild and deploy your repository's site. Boom you have your site up and running.
>
> Hopefully this works for you all. Shoutout to **@TheOdinProject**.

Thanks.

It kinda works, but all the images i have on the website just gone now :(

zooks commented on Oct 29, 2022

If you got this error:

```
 ! [rejected]        * -> gh-pages (non-fast-forward)
error: failed to push some refs to 'https://github.com/*'
hint: Updates were rejected because a pushed branch tip is behind its remote
hint: counterpart. Check out this branch and 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.
```

Just delete the branch `gh-pages` by this command:

`git push origin --delete gh-pages`

marc714 commented on Dec 25, 2022

This won't work on github pages:

```
const myPicture = new Image();
myPicture.src = "./src/assets/image.png"
```

You must use:

```
import myPicture from "./src/assets/image.png"
const banner = new Image();
banner.src = myPicture;
```

Make changes to your main branch (your gh-pages branch wont see changes):
`git push`

Make changes to your gh-pages branch:
`git subtree push --prefix dist origin gh-pages`

To refresh your chrome browser to see changes asap, press F12 to go into developer tools. Then right click on the reload icon next to your URL bar. Click 'Empty Cache and Hard Reload'

MrGassner commented on Feb 21

> If you already have a "gh-pages" branch, use the 1st command below. If you don't have a "gh-pages" branch, initialize it by using the 2nd command below. Make it easier to run by creating scripts like these in your package.json file:
>
> ```
> "scripts": {
>       "gh-deploy": "git push origin :gh-pages && git subtree push --prefix dist origin gh-pages"
>       "gh-deploy-init": "git push origin && git subtree push --prefix dist origin gh-pages",
> }
> ```
>
> In terminal:
>
> ```
> npm run gh-deploy
> npm run gh-deploy-init
> ```
>
> We aren't done just yet. Now go to your repository in GitHub. Go to Settings. Go to Pages. Under the Source section you will see a dropdown list of branches. Select the "gh-pages" branch and select the root as your folder. Then click Save.
>
> Give GitHub some time to rebuild and deploy your repository's site. Boom you have your site up and running.
>
> Hopefully this works for you all. Shoutout to **@TheOdinProject**.

This is what worked for me on the restaurant project of **@TheOdinProject**

Stan-Stani commented on Mar 7

> If you got this error:

> ```
>  ! [rejected]        * -> gh-pages (non-fast-forward)
> error: failed to push some refs to 'https://github.com/*'
> hint: Updates were rejected because a pushed branch tip is behind its remote
> hint: counterpart. Check out this branch and 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.
> ```
>
> Just delete the branch `gh-pages` by this command:
>
> ```
> git push origin --delete gh-pages
> ```

Thank you!

Stan-Stani commented on Mar 8

> If you got this error:
>
> ```
>  ! [rejected]        * -> gh-pages (non-fast-forward)
> error: failed to push some refs to 'https://github.com/*'
> hint: Updates were rejected because a pushed branch tip is behind its remote
> hint: counterpart. Check out this branch and 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.
> ```
>
> Just delete the branch `gh-pages` by this command:
>
> ```
> git push origin --delete gh-pages
> ```

Thank you!

> Does anyone know if this tutorial works if you are not dual-booting a linux OS? Say if i have created this project in a windows directory would this still work as intended? Having some issues figuring this out, as I cannot seem to find a good solution on how to publish this to gh-pages, nor if I was to publish it to another host i.e. Netlify or similar.

I have gotten it to work on Windows but I'm not using Yeoman.

Heechem commented on Mar 22

Nothing worked for me , im still receiving subtree isnt a git command , im unsing linux

rahhaly commented on Mar 25

> Nothing worked for me , im still receiving subtree isnt a git command , im unsing linux

Me too, I use Fedora.

tonybalde commented on Apr 5

I don't understand yet! I've tried but didn't work! I will search in google!

typenoob commented on May 23

can add it into npm run command

```
"scripts": {
  "deploy":"git subtree push --prefix dist origin gh-pages"
},
```

scottwright-dev commented on Jun 24 • edited

> git push origin :gh-pages && git subtree push --prefix dist origin gh-pages

Thank you for this **@fabriziomartini**, this is what helped me deploy my project.