How I update the GitHub pages for my projects with Sphinx documentation

Author: Pete R. Jemian

url: http://prjemian.github.io

this: http://prjemian.github.io/gh-pages.html

pdf: http://prjemian.github.io/gh-pages.pdf

see: https://help.github.com/articles/creating-project-pages-manually

Here are the steps I follow (only slight difference from github's):

1. build the Sphinx docs in the project locally
2. copy the *html* directory contents somewhere else
3. switch to the *gh-pages* branch
4. remove all git content
5. copy contents of that *html* dir back to the branch
6. commit all the new files
7. push to github
8. switch back to the *master* branch

build the Sphinx docs in the project locally

for my projects, the docs are in the "docs" subdirectory:

```
$ cd docs
$ make clean
$ make html
```

copy the *html* directory contents somewhere else

I use *tar*, it is simple (proceeding from previous point):

```
$ cd build/html
$ tar czf /tmp/html.tgz .
```

switch to the *gh-pages* branch

first, move back to the project's root directory:

```
$ cd ../../..
```

If the project does not have a *gh-pages* branch, it should be created with the special command:

```
$ git checkout --orphan gh-pages
    Switched to a new branch 'gh-pages'
```

You only need to do this once per project. Check https://github.com/prjemian/cmd_response [1] under branches and it will tell you if such a branch exists. For *cmd_response*, that exists. If the *gh-pages* branch exists, just use this command:

```
$ git checkout gh-pages
    Branch gh-pages set up to track remote branch gh-pages from origin.
```

remove all git content

Confirm you are on the gh-pages branch if you need the confidence builder:

```
$ git branch
* gh-pages
  master
```

The "*" confirms the *gh-pages* branch is checked out now.

This next command looks dangerous. Fear not. It just cleans out the project content from the documentation branch:

```
$ git rm -rf .
rm '.buildinfo'
rm '.nojekyll'
rm '_modules/index.html'
    ...
```

All that should remain is the *.git* directory. Don't delete that!

At this point, there may remain some other files and directories that were not in git version control. These need to be deleted directly (not with git but with normal delete commands). Check for them. Likely ones include docs, dist, build, perhaps others. For me:

```
$ ls -lAFg
total 12
drwxr-xr-x 3 mint14 4096 Mar 24 20:30 docs/
drwxr-xr-x 8 mint14 4096 Mar 24 20:30 .git/
    $ /bin/rm -rf docs
    $ ls -lAFg
    total 4
    drwxr-xr-x 8 mint14 4096 Mar 24 20:30 .git/
```

All that should remain *now* is the *.git* directory.

copy contents of that html dir back to the branch

We used tar before to copy our documentation. We bring it back now:

    $ tar xzf /tmp/html.tgz

commit all the new files

Put all the new documentation into git version control:

```
$ git add .
    $ git commit -a -m "publish the docs"
```

push to github

The changes are not published until you push the changeset back to github:

```
$ git push origin gh-pages
```

and enter credentials as requested. Your documentation should appear at http://prjemian.github.io/cmd_response right away if they have already been posted before. For a brand new project, it might take up to 10 minutes.

switch back to the *master* branch

Don't forget to switch your working directory back to the *master* (or other) branch once you have successfully pushed the docs:

```
$ git checkout master
```

---

1 the general pattern for the project URL is: https://github.com/<username>/<projectname>

The docs will be published to: http://<username>.github.io/<projectname>