

# How to Create and Launch a Website from Start to Finish

A beginners guide to publishing websites using themes from Start Bootstrap and GitHub Pages

August 19, 2019 by [David Miller](#)

[Guide](#)

[Beginner](#)

---

GitHub Pages is a great way to get a simple, static website published online - especially when you use a predesigned theme from Start Bootstrap!

There are plenty of benefits to having your code on GitHub, but for the purposes of this guide, the major bonus is that you can host a static website for free! So let's get started!

## Before We Begin

Technical Requirements: In this guide you will be editing the content of a website theme! You will need to have some working knowledge of HTML to do so.

Optional Requirements: If you want to edit your theme's style (colors, fonts, etc.), you will need to have some working knowledge of CSS - this isn't required if you are happy with the default design of the theme you choose.

## Step 1: Create a GitHub Account

---

First things first, since we will be setting up a website using GitHub Pages, you will need a GitHub account. If you don't already have one, just go to <https://github.com> and create an account - it's free!

## Step 2: Download GitHub Desktop

---

It's common for developers to use their terminal to push and pull files to and from GitHub, but since this is a beginner level guide we're going to be using GitHub's super handy desktop client instead.

GitHub Desktop is also free to download, and it's available at <https://desktop.github.com>. After downloading and installing, open up GitHub Desktop and login using your GitHub account credentials.

## Step 3: Create a New Repository

After you have GitHub Desktop installed, you will create a new repository. For now, it will be blank. We will choose a theme in the next step!

Note: A *repository* is just a fancy way of saying a place where code and files are stored!

After opening GitHub Desktop, go to **File > New Repository**. The dialog box that opens will allow you to enter some details. Give your repository a name and a description. Most importantly, you will need to set the *Local Path*. Select the **Choose...** option and point to a folder where you will want your website files to remain!

Don't worry about the README, Git Ignore, or the License options. After you're done, select **Create Repository**.

Create New Repository on GitHub Desktop

*Make sure the Local Path is where you want your website files to be!*

## Step 4: Choose a Theme from Start Bootstrap

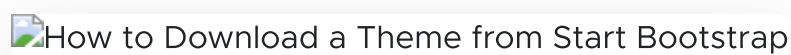
---

Now that we have a repository created, it's time to choose a theme! You can browse the predesigned Bootstrap themes at <https://startbootstrap.com/themes> to get started. Make sure to choose an HTML theme or template, not one of our Jekyll or Angular projects. These are more advanced level systems, and will not work with this guide!

## Step 5: Download Your Theme

---

After you've found a theme you like, click on the **Free Download** button that is located on the right side of the overview page. You can see an example below from the overview page of our Resume theme.

How to Download a Theme from Start Bootstrap

*Click the Free Download button to download a theme from Start Bootstrap*

## Step 6: Unzip Your Downloaded Theme

---

This one is a pretty simple step. The theme you downloaded comes as a .zip file. Unzip the file, and you should be able to look inside the extracted file and see the HTML, CSS, and other supporting assets necessary to use the theme. Remember where these files are, we will be moving them in next step.

## Step 7: Move the Theme Files into your New Repository

---

The repository you created in Step 3 should be a blank folder. Take the files that you extracted in Step 6 and move the files into the repository you created.

**Note:** Don't move the whole folder, move the files from *inside* the folder into the blank folder that you created in Step 3. The `index.html` file should be in the root level of the repository.

## Step 8: Edit the Theme Files

---

After your theme files are in your new repository, it's time to edit the files themselves! Open the `index.html`, or any other HTML files, in your favorite code editor and make any changes that you need to make!


Once you're finished adding your content to the HTML files, you can also make adjustments to the CSS if that is something you want to do!

## Step 9: Commit and Push the Modified Files

---


Now open the GitHub Desktop application again. Since you edited some files locally on your computer, the program will detect the files that have changed.

It's time to push these changes to GitHub! Simply enter a commit message in the box provided (in the lower left hand part of the application), and a description if you want to, and then click on **Commit to master** - shown below.



Summary (required)

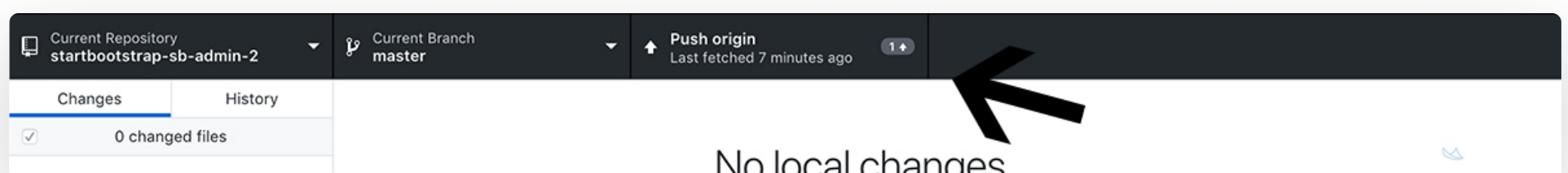
Description



Commit to master

Enter a commit message in the box shown and commit the changes

This doesn't push your changed files GitHub yet! To do that you will need to click on **Push origin** option at the top of the screen - shown below.



Enter a commit message in the box shown and commit the changes

Now, if you go to your repository on GitHub, you will see that the changes have been made! Go to your repository on GitHub's website and verify that the changes were made to the repository. On your repository, you should see the commit message with a timestamp on the files that have changed.

## Step 10: Set GitHub Pages Settings

On your repository on GitHub, got to Settings and scroll down to GitHub Pages. Make sure you have publishing to GitHub Pages turned on. The file should be set to /root since we will be putting the files in the root director.

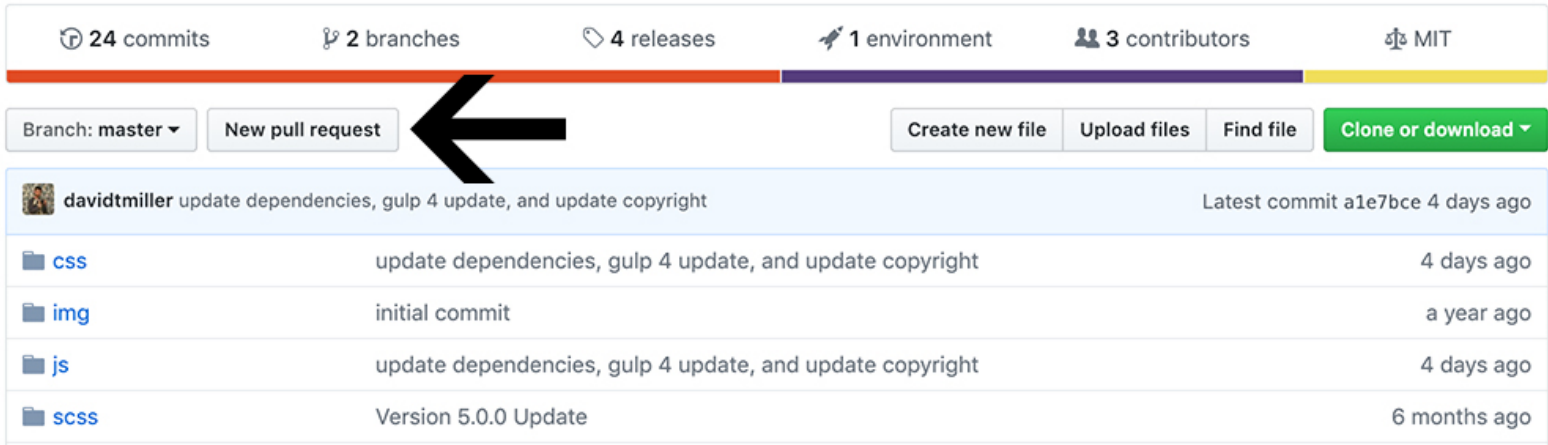
## Step 11: Merge the Files Into the gh-pages Branch

Your changes have been committed and pushed to the master branch of your repository, but when using GitHub pages, the files in the gh-pages branch are the ones that are hosted. That means we need to merge the modified files from the master branch into a gh-pages branch!

If you don't have a gh-pages branch, you can make one by selecting the Branch dropdown and creating a new branch called **gh-pages**.

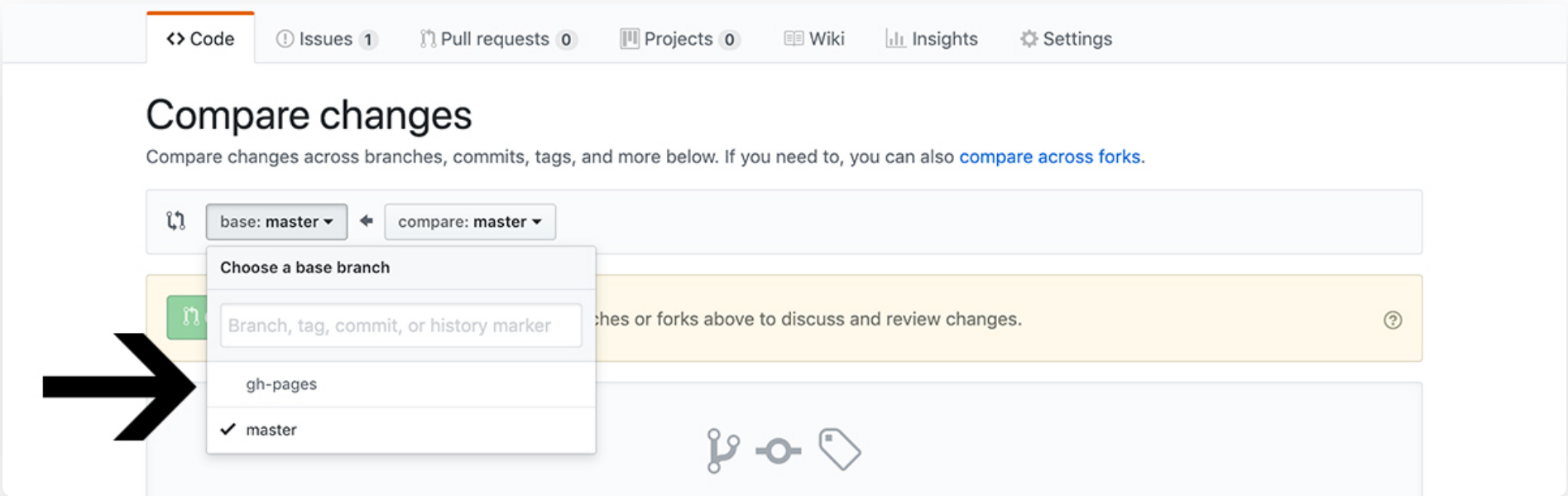
Once you've done that, you are going to make what is called a *pull request*.

Merging your master branch into gh-pages takes a few steps. First you will need to go to your repo and click on the 'New pull request' button - make sure your master branch is selected!



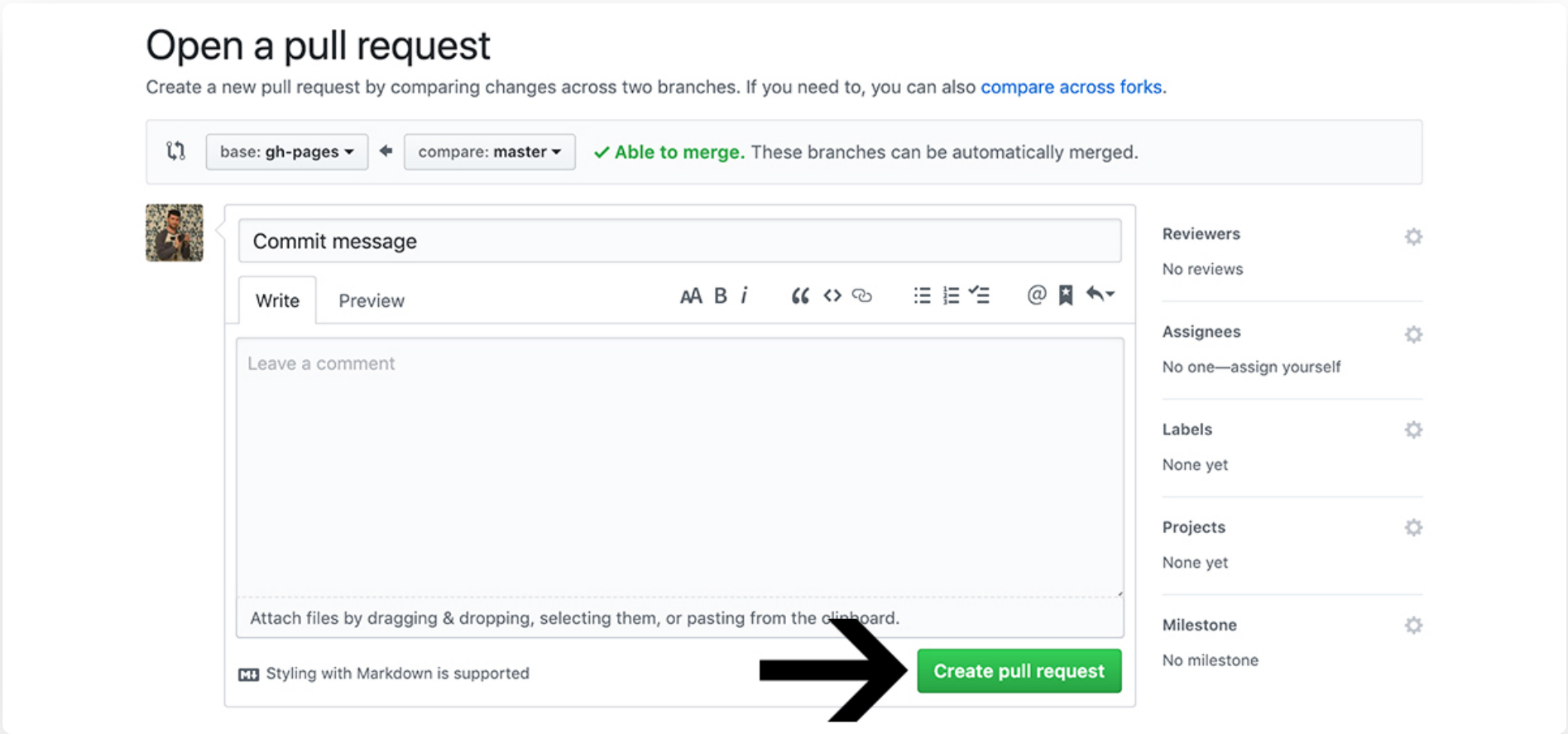
Click on the pull request tab and make a new pull request

Next, you will need to make sure you're comparing the changes made in the master branch to the gh-pages branch. That is where we want the files to go. Change the base repository in the dropdown provided to gh-pages.



Make sure you are merging into the gh-pages branch!


At this point, you should see the commit message and a summary of the files that will be merged. Click on **Create pull request** - seen below.



Open the Pull request

Finally, you will need to click on **Merge pull request** and then **Confirm merge** to finalize the merge.





✓

**This branch has no conflicts with the base branch**  
Merging can be performed automatically.

Merge pull request

▼

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Merge the Pull request

If everything has gone as planned, the gh-pages branch will now be identical to the master branch, and GitHub Pages will be serving your files to a website located at `username.github.io/repository-name`. Keep in mind that it will probably take a few minutes for the changes to go live as the server catches up. Give it some time, and check to make sure your website is live after a few minutes!


### Additional Details

If you want to update your website, just use this same guide and repeat steps 8 and beyond. You'll be making changes locally, pushing the changes to the master branch on your GitHub repository, and then merging the master branch into the gh-pages branch again!

If you want to use a custom domain for your website, you can also set that up with GitHub pages. You will need to purchase a domain from a domain name provider, and then point the domain to GitHub's name servers. GitHub has some handy documentation on how to do this on their custom domain setup page!


That's it! You now have a website online with free hosting thanks to GitHub pages! Thanks for checking out this Start Bootstrap beginner tutorial!


This guide was updated on **October 2nd, 2020**.




David Miller

David Miller is the creator and founder of Start Bootstrap. He is a front end web designer working out of Orlando, Florida.


 [davidmiller.io](#)

 [davidmillerskt](#)

 [davidtmiller](#)


24 Comments


Start Bootstrap


 Disqus' Privacy Policy

1


 Login ▼

 Recommend 5

 Tweet


 Share

Sort by Best ▼




Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 


Name



Mohammad Mn • 2 years ago

the article's title should read: How to Create and Launch a Website from Start to Finish in Minutes :)

2 ^ | ▼ • Reply • Share ›

 **Start Bootstrap** Mod ➔ Mohammad Mn • 2 years ago

Get a domain and create a website with Squarespace. Start your free trial today.

ads via Carbon



Recent Guides

[How to Create and Launch a Website from Start to Finish](#)  
[August 19, 2019](#)

[View All Guides](#)

Want more Bootstrap themes, templates, and UI tools?  
Subscribe to get updated when new Bootstrap related products arrive!

Email address...

Subscribe!

*This site is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms of Service](#) apply.*



Start Bootstrap

Free & premium UI tools to help you build better websites



PRODUCTS

- [Themes](#)
- [Templates](#)
- [Snippets](#)
- [Guides](#)

SUPPORT

- [Knowledge Base](#)
- [Custom Development](#)

RESOURCES

- [Blog](#)
- [Bootstrap Resources](#)
- [SB Angular Inspector](#)

COMPANY

- [About Us](#)
- [Privacy Policy](#)
- [Terms and Conditions](#)
- [Licenses](#)