2.6.7 Deploy

The new version of the landing page is ready to go live! At this point, we could merge the develop branch directly into the master branch, but one of Run Buddy's internal developers would like to do a code comparison and review first.

ON THE JOB

Code reviews are an important part of the development process. No code should go into production before it's been reviewed by other members of the team. This is a great way to enforce best practices among developers, fix overlooked syntax errors, and catch potential conflicts like a developer renaming a CSS class that another developer was using.

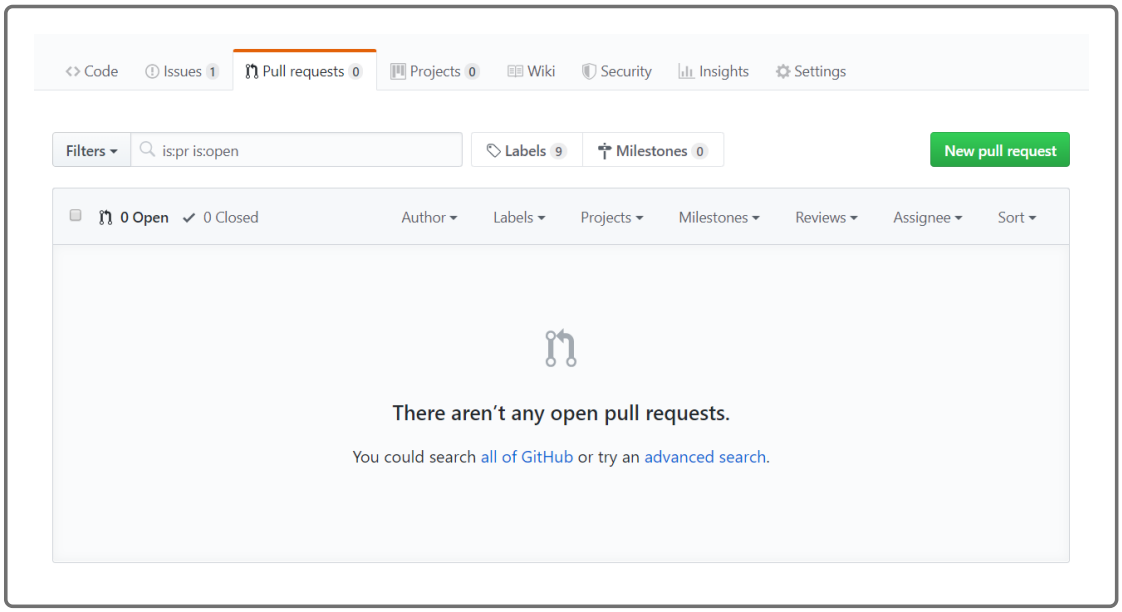
Fortunately, GitHub provides an easy mechanism for code reviews. Branches can be merged through GitHub's interface via a process called **pull requests** (or **PR**), and every pull request can be reviewed by other team members before the merge actually happens. A more appropriate term might be "merge request," but pull request has become standard

language that essentially means a request for someone else to "pull" your branch into their repository.

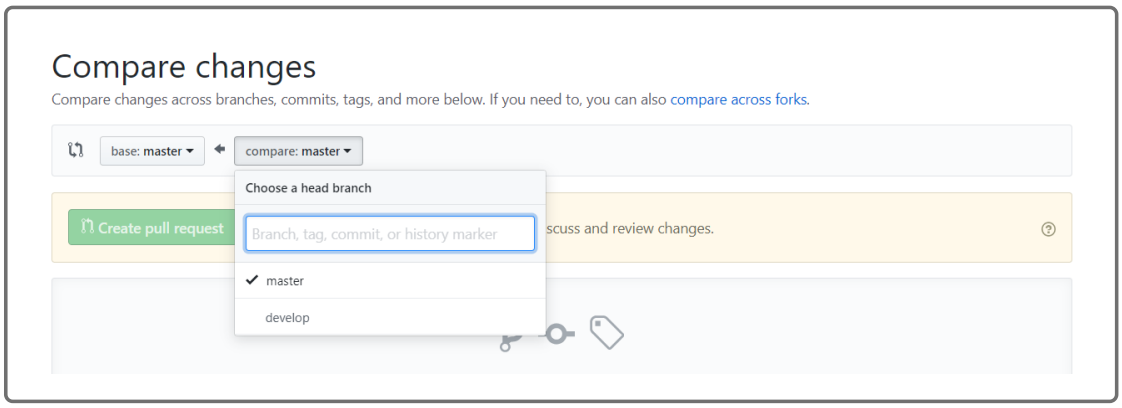
REWIND

Think back to when you made a README file on GitHub and used the command `git pull` to download that commit into your local repository.

Let's go through the pull request process for Run Buddy. Navigate to your GitHub repository in the browser and click the "Pull requests" tab. This should route you to the following page:

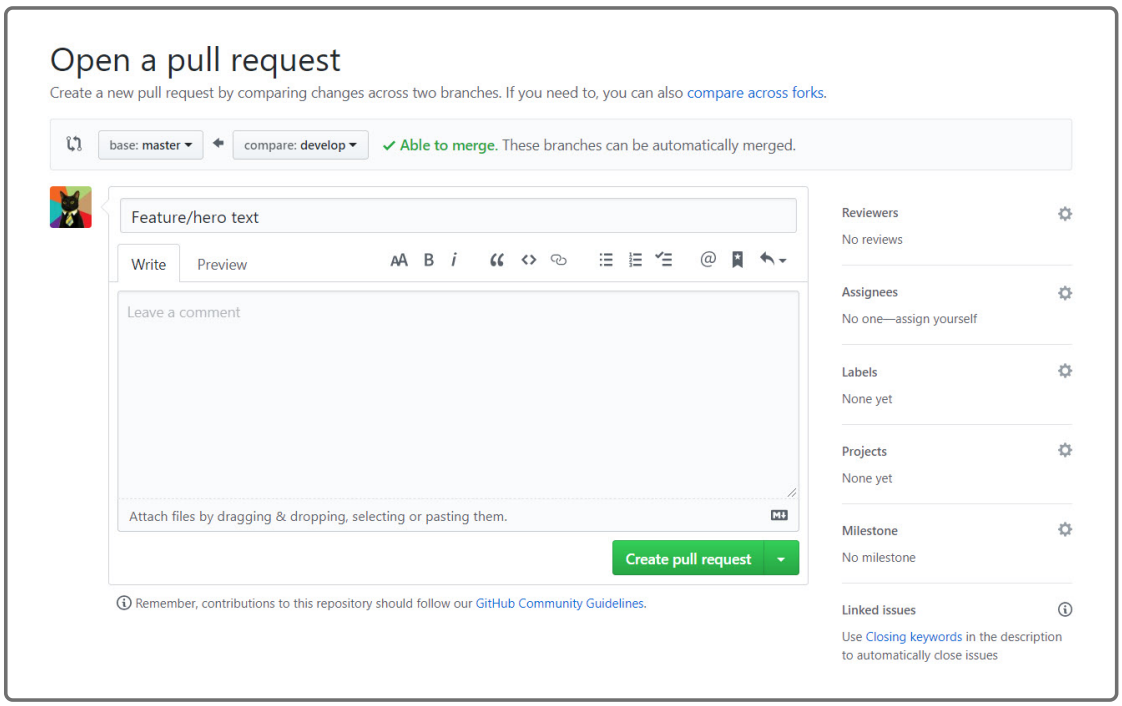


Click on the green "New pull request" button on the right. This will route you to yet another page where you'll have the option to choose and compare which branches are being merged:



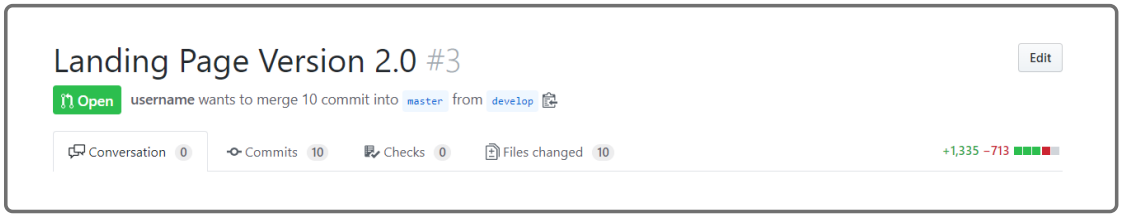
The "base" dropdown defaults to `master`, which is fine. That's what we want. The second dropdown (compare) is where we choose the branch that will be merged into `master`. Click the dropdown and select `develop`.

After doing so, you should now be able to click the "Create pull request" button, which will route you to a new page:



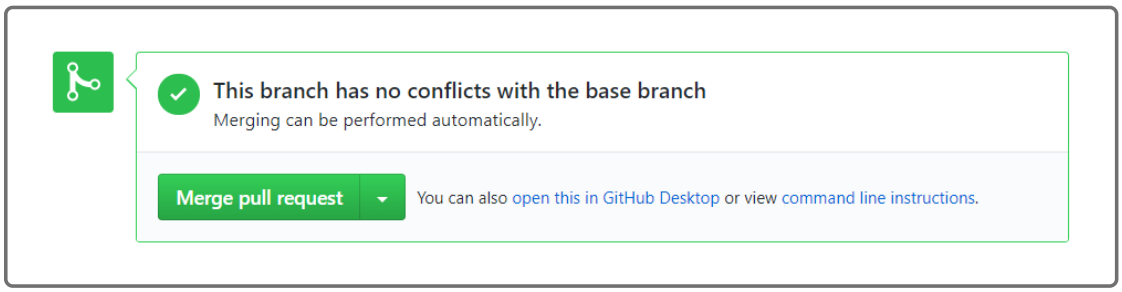
On this page, GitHub gives you the option to enter any comments before the pull request is created. The name of the pull request defaults to the last commit message that was made (in this case, "hover animation"). Change the name to something more descriptive, like "Landing Page Version 2.0." Then click the "Create pull request" button.

The next page is the actual pull request page that includes an overview of what this pull request aims to do:



Click the "Files changed" tab to see a breakdown of every file and line of code that's different between `master` and `develop`:

This is where the actual code review would take place. The Run Buddy developer can leave comments here at specific line numbers where there might be a problem. If there are no problems that need to be addressed, the developer can return to the previous PR overview page and click the green "Merge pull request" button:



This will complete the merge and apply all of the commits from `develop` onto `master`. In our case, the Run Buddy developer gave us their blessing but never merged the PR. Let's go ahead and click the button to merge it ourselves. Because we set up the GitHub Page ahead of time, our live webpage should now reflect Run Buddy version 2. Visit `https://<username>.github.io/run-buddy/` (replacing `<username>` with your GitHub username) to see if it worked!

IMPORTANT

> Sometimes a webpage still shows an older version of itself even after pushing new changes. This is because the browser has **cached**, or saved, the older versions of the HTML and CSS files to make the page load faster on subsequent visits. You can perform a **hard refresh** to clear the browser's cache for a specific page by holding down Ctrl on Windows or Shift on Mac and clicking the reload button.
>
> Whether that was the problem or not, know that many developers run into similar problems all the time. A quick **Google search for "GitHub Pages not updating" (https://www.google.com/search?q=github+pages+not+updating)** will reveal that just as many developers are also willing to answer and help troubleshoot these questions!

Now that the new Runny Buddy site is finished and deployed, use the following knowledge check to assess your understanding of this lesson's takeaways:

What does the selector `div:hover a` apply to?

- ◯ Any `<div>` element whose child `<a>` element is in a hover state.
- ◯ Any `<a>` element inside a `<div>` that's in a hover state.
- ◯ Any `<div>` or `<a>` element that's in a hover state.
- ◯ Any `<a>` element sitting next to a `<div>` that's in a hover state.

Check Answer

What do the values in `transition: color 1s` mean?

- ◯ Apply the transition to the `color` property that lasts one second.
- ◯ Transition the color from its original value to plus one on the RGB scale.
- ◯ Apply the transition to the `color` and `1s` properties.
- ◯ Apply the transition to the `color` property, but only run once.

Check Answer

What is the purpose of `overflow: hidden` ?

- ◯ Hide all content inside of the element.
- ◯ Hide any neighboring content.
- ◯ Hide content that flows outside of the element.
- ◯ Fix issues caused by CSS floats.

Check Answer

Finish ▸