



How to Review a Pull Request

HFLA WEBSITE TEAM

<https://github.com/hackforla/website/wiki/How-to-Review-Pull-Requests#Intro>

What is reviewing a Pull Request?

As a reviewer you want to ensure the ABCs -
changes are:

- Applicable
- Don't **Break** the website
- Clean

approve a PR if the ABCs are satisfied,
otherwise, **request changes**

- Leave a comment either way

The reviewer checks the collaborator's changes

You can approve a PR or request changes

It's important to leave feedback whether you approve or request changes

See [leaving feedback](#)



Find Pull Requests: Step -1

Order for reviewing Pull Requests if you're new to it

- **Good First Issue**
- **Good Second Issue**
- **Medium**
- **Large**

THERE'S NO WRONG WAY TO PICK A PR ~ JUST TRY YOUR BEST :)

Add yourself as a reviewer

- Use cog on Right side to add yourself as a reviewer
- Comment PR review availability and ETA
- Make sure the correct Project Board is selected
- If the correct branch is selected and the issue is properly linked:
- Copy branch info for your Code Editor
- Paste into your local HfLA repo*

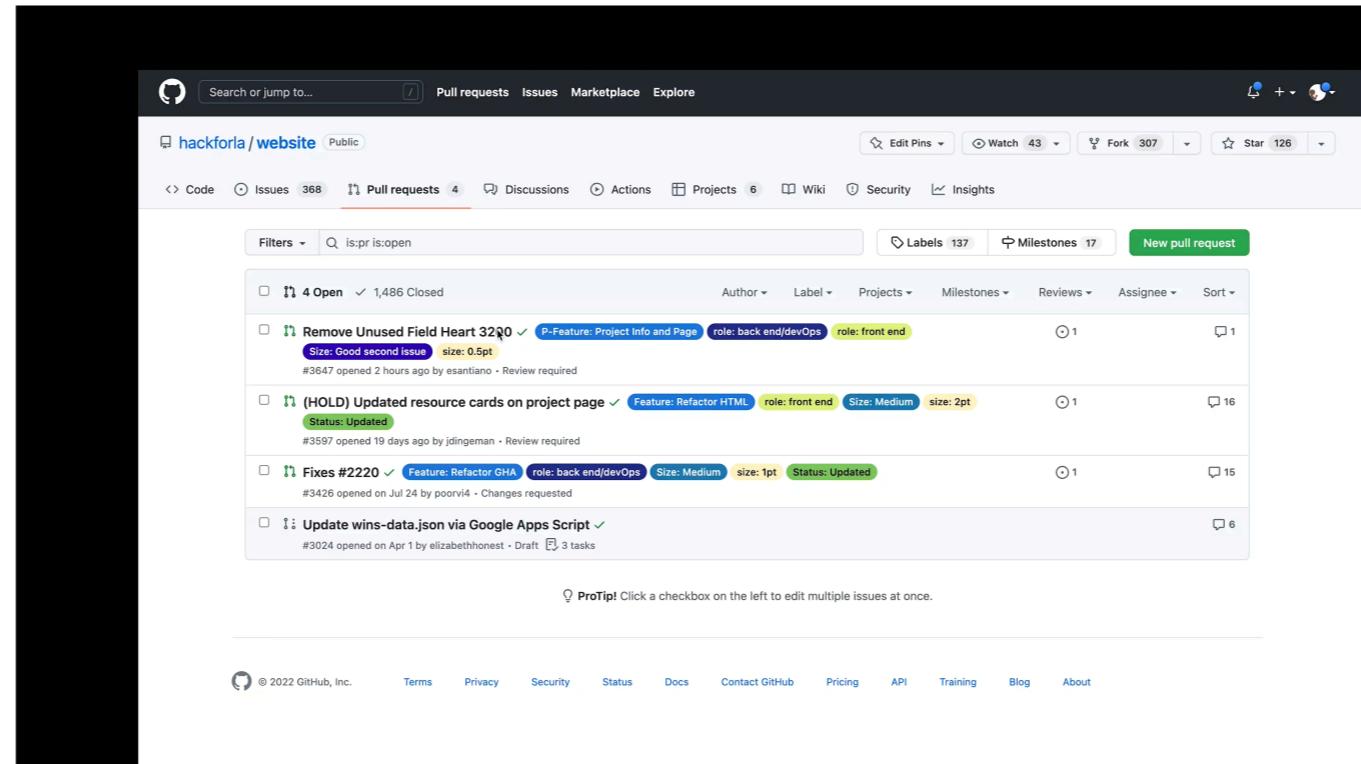
* Best practice: refresh your forked repo and local repo before reviewing a PR

It's okay to review a PR that someone else is reviewing!

Review ETA: 6 PM 3/4/22 / Availability: 5-8 PM Monday

If the project board is not selected, ask the collaborator to put it on the board when you make your review

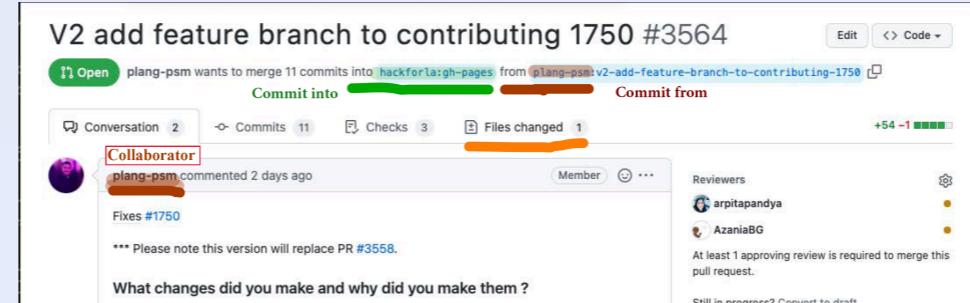
It's a good idea to refresh your forked and local repo before downloading the collaborator's branch



STEP 0

CORRECT BRANCH?

Request changes if not



Check that the PR has the correct branch:

1. Commit into = **hackforlagh-pages**
2. Commit from = **collaborator (plang-psm)**

The commit into must be **hackforlagh-pages**

The commit from must be from the collaborator who opened the PR

Request changes and leave a comment asking that the PR be made with the correct branch

STEP 1

LINKED ISSUE?

- Request changes if missing
- Screenshots are added to the PR (if applicable)?
- Check for properly linked issue, e.g.:
Fixes #1326
Resolves #1326
Closes #144

Aveline-art commented 13 days ago • edited

Fixes #1326
Adjusted scss file according to #1326 for the purpose of matching Figma design
Community feature
Figma design reference
Screenshots of updated site, below, in various resolutions
@daniellex0 Is this acceptable or would you like additional adjustments beyond the actions items in #1326 to better match the Figma design?

Mobile Android
GalaxyS9 vertical (360x740)

Communities of Practice

Communities of Practice started at Hack for LA in 2020, post-Covid. These communities have become a key connection point in our fully remote organization. They have enabled us to connect with individuals across the globe that are both meaningful for these individuals, and have helped the organization to build a robust infrastructure to support our community, operations and strategic projects. These communities have also enabled Hack for LA to provide rapid response to the evolving crises of 2020.

Reviewers: daniellex0, erikaBell
Assignees: No one - assign yourself
Labels: None yet
Projects: None yet
Milestone: No milestone
Linked issues: Successfully merging this pull request may close these issues.
Standardize Communities of Practice page h...

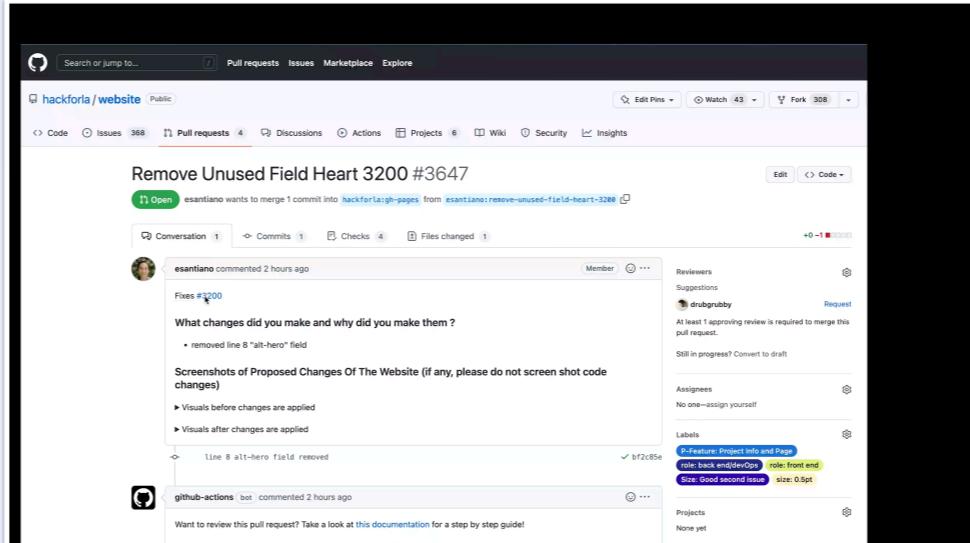
Leave a comment and request changes if the linked issue is missing or if before / after screenshots are missing

If linked issue is not in the proper format, proceed with the review but add a comment on how to properly link the issue

STEP 2

UNDERSTAND LINKED ISSUE

- Visit the issue
- Research unfamiliar terms
- Ask questions
- Request review with a teammate if unsure



Visit the issue in order to properly assess the collaborator's changes

Don't just blindly review changes without a firm understanding

If you don't understand the issue, that's perfectly normal for a number of reasons, so ask questions and/or research unfamiliar terms

STEP 3

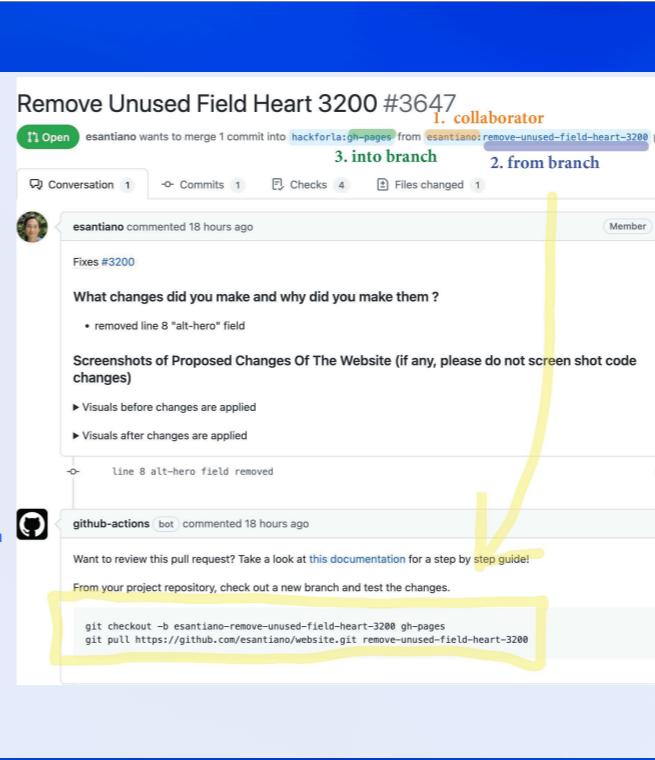
DOWNLOAD BRANCH

To download changes, need:

1. **Name of collaborator** - GitHub handle of PR person
2. **Name of from branch** - name of collaborator's branch
3. **Name of into branch** - name of website's repo branch

```
git checkout -b NameOfCollaborator-nameOfFromBranch NameOfIntoBranch
```

```
git pull https://github.com/NameOfCollaborator/website.git nameOfFromBranch
```



Instructions to download collaborator's branch are in Step 3 of the guide (Development Team), but they may also be found in the PR
Step 1: create a new branch based on the collaborator's PR - name and “from” branch space name-of-into-branch

Step 2: pull those changes

Step 3: Start project with docker and view changes in browser at localhost:4000

STEP 3

(CONTINUED)

VIEW CHANGES IN BROWSER

- Start project locally
- Navigate to <http://localhost:4000>
- Locate page(s)* with changes
- Compare changes to the live HfLA website
- View changes in multiple viewports (mobile, tablet, etc.) using developer mode
 - ◎ [Google Chrome](#)
 - ◎ [Mozilla Firefox](#)
 - ◎ [Microsoft Edge](#)



Check for the following:

- Are the changes applicable to the issue?
- Do changes go beyond what the issue requires?
- Does the website appear less user-friendly?
- Do links and components still work as intended?
- Ask the ABCs

* If there are CSS changes, check multiple pages; not just the target page

Step 3: Start project with docker and view changes in browser at localhost:4000

If there are CSS changes made, check multiple pages; not just the target page (clear metadata if changes are not apparent)

Step 4: Compare to HfLA website and check for the following...

Step 5: Request changes if there is an issue

STEP 4

LOOK AT THE SOURCE CODE

- Click on the [Files changed](#) tab
- Green highlights or + sign are code additions
- Red highlights or - sign are code deletions
- Make sure to check all files in [Files changed](#) tab

```
@@ -11,7 +11,7 @@
11 11     {% elseif name == 'Getting Started' %}
12 12         {% include svg/logo-hfia-small.svg %}
13 13     {% elseif image %}
14 14     -         
15 15     +         
16 16     {% else %}
17 17     {% endif %}
```

If the ABCs are satisfied, look at the source code by clicking on the [Files Changed](#) tab

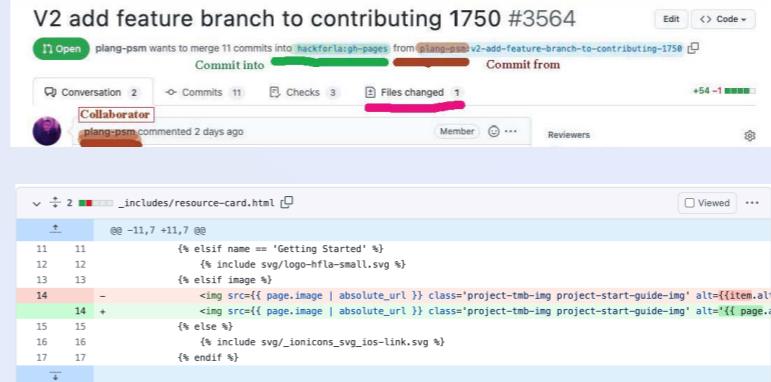
Green highlights or lines with a plus sign = additions to code

Red or lines with minus sign = deletions

STEP 4 (CONTINUED)

LOOK AT THE SOURCE CODE

- Can the changes be condensed?
- Are there drastic changes that do not belong?
- Are there changes missing?
- Leave an inline comment and request changes if there is a problem



```
diff --git a/_includes/resource-card.html b/_includes/resource-card.html
@@ -11,7 +11,7 @@
11 11      (% elseif name == 'Getting Started' %)
12 12          (% include svg/logo-hfia-small.svg %)
13 13      (% elsif image %)
14 14      -      <img src={{ page.image | absolute_url }} class='project-tmb-img project-start-guide-img' alt='{{item.alt}}'
+      +      <img src={{ page.image | absolute_url }} class='project-tmb-img project-start-guide-img' alt='{{ page.item.alt }}'
15 15      (% else %)
16 16          (% include svg/ionicons_svg_ios-link.svg %)
17 17      (% endif %)
```

Look at the source code by clicking on the Files Changed tab and check for the following...

-> leave an inline comment and request changes if there is a problem

STEP 5

CHECK FOR ANYTHING ELSE

- Do changes cause an unforeseen problem?
- Are the changes inadequate, erroneous, or incomplete?
- Was there a problem with the original issue?
- Is there a problem with the file structure?

Discuss with the creator of the issue if any of these apply

Check for anything else

- sometimes files aren't where they used to be or there are problems with file structure (can check local repo)

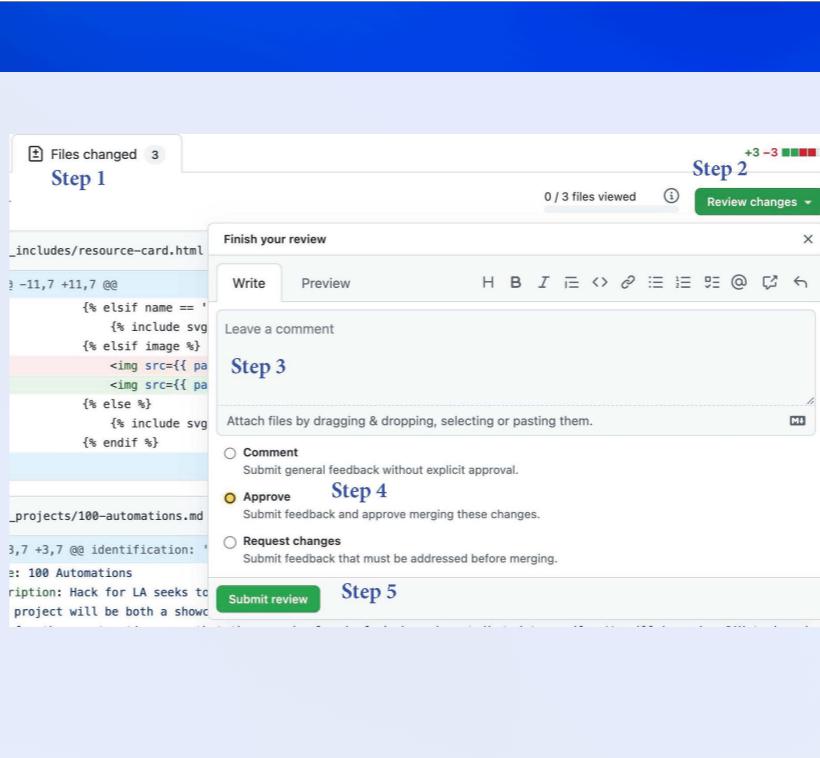
If there are other problems, leave a comment to discuss with the creator of the issue

You can also add the creator of the issue as reviewer so they can review the PR with you

STEP 6

APPROVE THE PR

1. Click the **Files changed** tab
2. Click the **Review changes** button
3. Add a comment with positive feedback
4. Select the **Approve** radio button
5. Click the **Submit review** button



If all checks are satisfied you can approve the PR with the following steps...



CHECKLIST

<https://github.com/hackforla/website/wiki/How-to-Review-Pull-Requests#Checklist>

- Does the PR have the correct branch?
- Is there a linked issue?
 - understand the linked issue
- View changes in the browser
- View files changed tab
- Check for anything else
- Approve the PR (or request changes)

RESOURCES

GUIDE:

<https://github.com/hackforla/website/wiki/How-to-Review-Pull-Requests>

Find pull requests to review

<https://github.com/hackforla/website/pulls>

Checklist:

<https://github.com/hackforla/website/wiki/How-to-Review-Pull-Requests#Checklist>

Properly link issue:

<https://docs.github.com/en/issues/tracking-your-work-with-issues/linking-a-pull-request-to-an-issue#linking-a-pull-request-to-an-issue-using-a-keyword>

Leave inline comment

<https://github.com/hackforla/website/wiki/How-to-Review-Pull-Requests#leave-an-inline-comment>