



# GitHub 102

by

Renana Yacobi

Annu Joshi

# Quick Recap

# Git vs GitHub

## Git

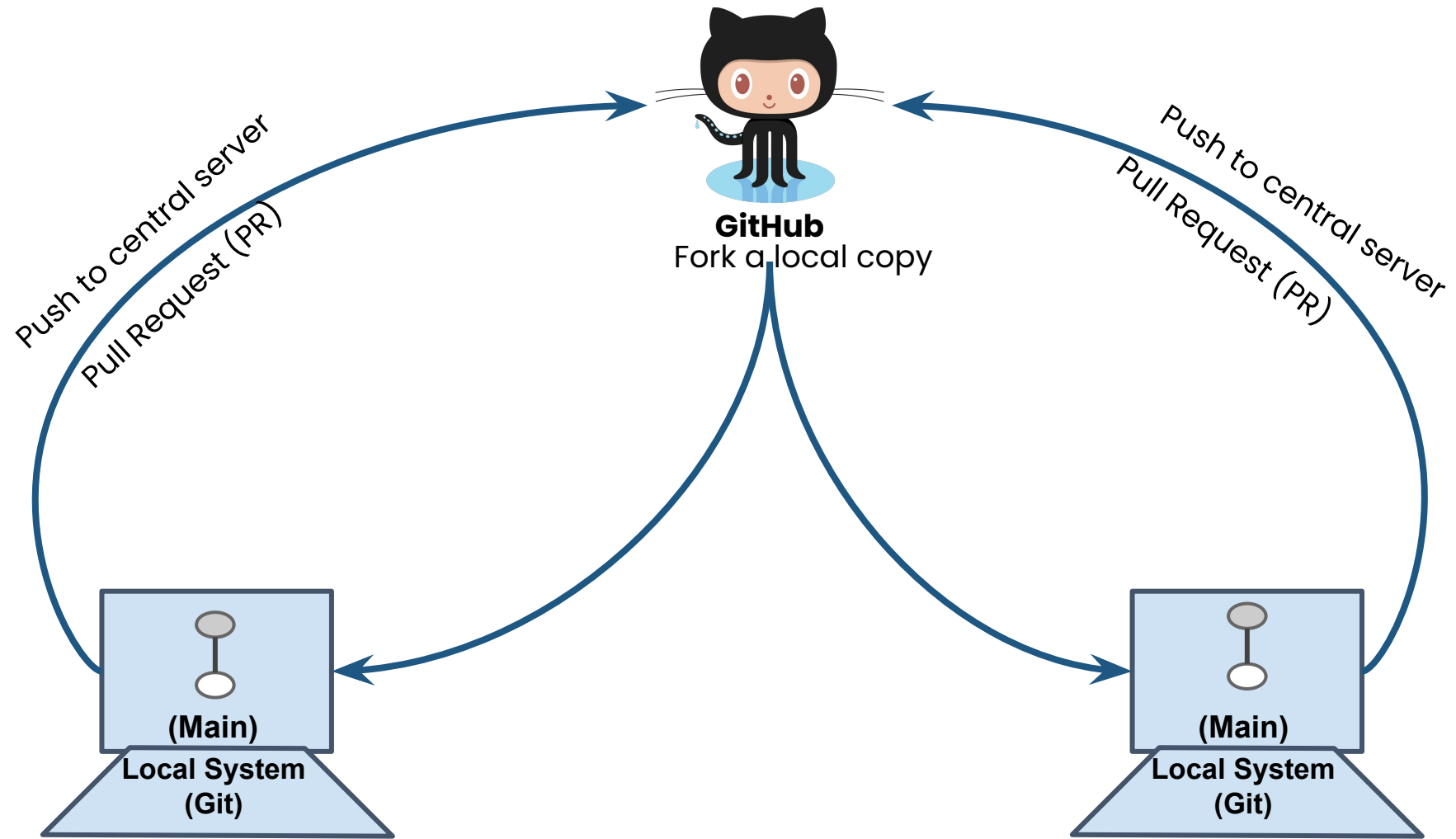


Git is a version control system that lets you manage and keep track of your source code history in your local system.

## GitHub



Cloud-based hosting service that lets you manage your Git repositories.





# What you should already know:

## Git 101

- git init
- git add
- git commit
- git clone
- git pull
- git push

## GitHub 101

- Fork
- Clone
- Origin
- Fetch and Merge
- Making changes
- Creating a Pull Request

# Branches

# Local branches

Let's create our work environment:

```
> mkdir git-102  
> cd git-102  
> git init  
> git branch  
>
```

# Local branches

'main' branch was not created yet, let's create it:

```
> echo echo Hello World > hello-world.sh
> git add .

> git commit -m "creating first branch"
[main (root-commit) a700914] creating first branch

1 file changed, 1 insertion(+)
create mode 100644 hello-world.sh

> git branch
* main
```



# 'main' branch

We usually start with branch 'main'.

This is our default branch.

```
> git status  
On branch main  
nothing to commit, working tree clean
```

# Creating new local branch

Let's create a new local branch.

Checkout command is the best option.

'-b' means create a new branch.

```
> git checkout -b my-first-branch  
Switched to a new branch 'my-first-branch'
```



# Congratulations!

You just created your second local branch.



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

# Edit local branch

Any commit will change only the branch:

```
> echo echo Hello World1 > hello-world.sh
> git add .
> git commit -m "Changed message to Hello World1"
[my-first-branch cc20e23] Changed message to Hello
World1
1 file changed, 1 insertion(+), 1 deletion(-)
```



# Switching between local branches

Checkout command without '-b' will switch between the branches.

```
> git checkout main  
Switched to branch 'main'
```

```
> git checkout my-first-branch  
Switched to branch 'my-first-branch'
```

# Create a remote repository

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

---

Owner \*

Repository name \*


renana ▾

/ github-102 ✓


Great repository names are short and memorable. Need inspiration? How about [congenial-tribble?](#)

Description (optional)

---

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

---

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more](#).

☐ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more](#).

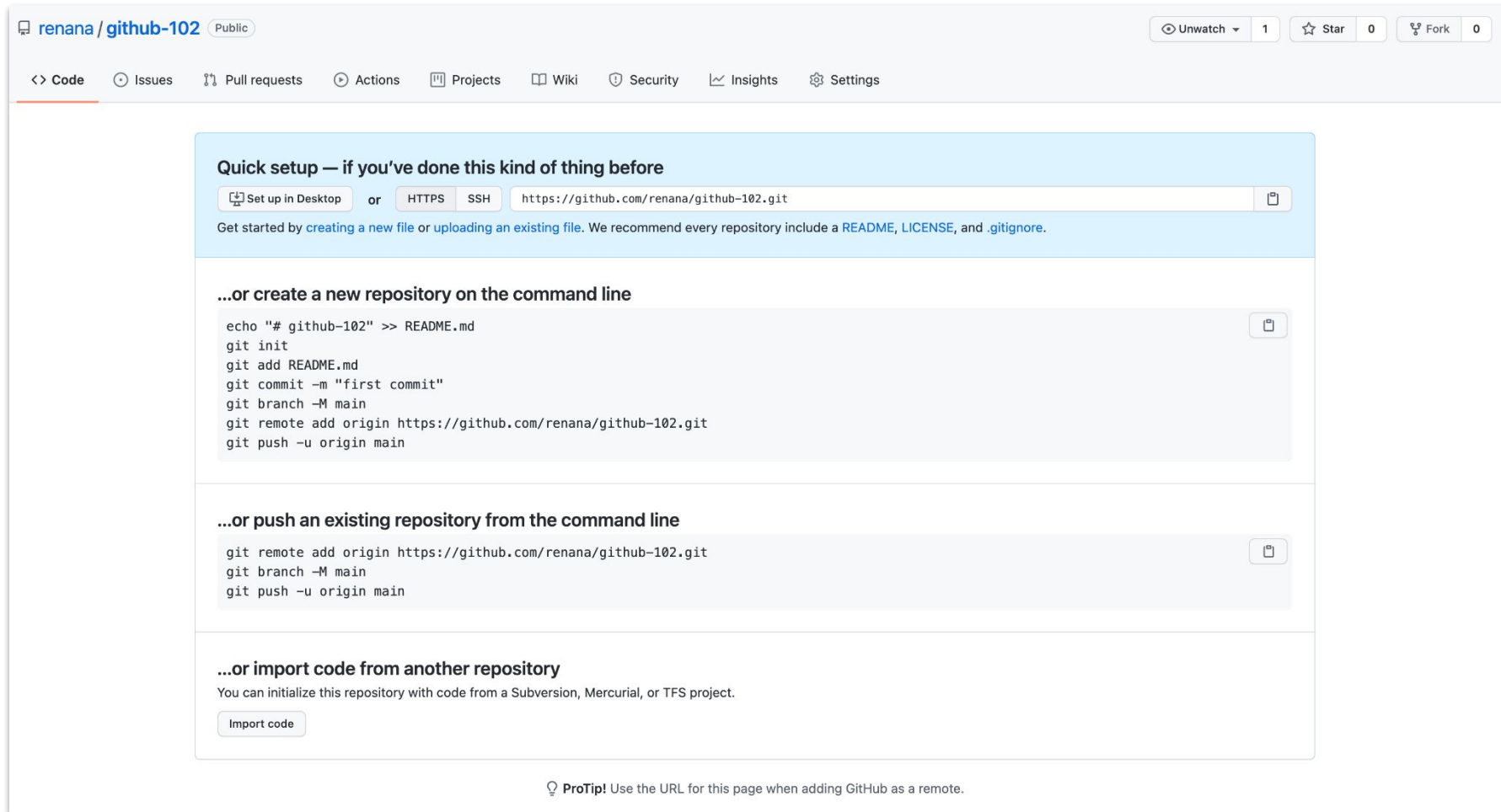
☐ **Choose a license**

A license tells others what they can and can't do with your code. [Learn more](#).

---

Create repository

# Your new remote repository



The screenshot shows the GitHub interface for a new repository named 'github-102' by user 'renana'. The repository is public. At the top, there are navigation tabs: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the repository name, there are buttons for 'Unwatch', 'Star' (1), and 'Fork' (0). The main content area provides instructions for setting up the repository. It starts with a 'Quick setup' section for users who have done this before, offering options to 'Set up in Desktop', 'HTTPS', or 'SSH'. The 'HTTPS' option is selected, showing the URL 'https://github.com/renana/github-102.git'. Below this, it says 'Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.' The next section is '...or create a new repository on the command line', which includes a code block with the following commands: 

```
echo "# github-102" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/renana/github-102.git
git push -u origin main
```

 The following section is '...or push an existing repository from the command line', which includes a code block with the following commands: 

```
git remote add origin https://github.com/renana/github-102.git
git branch -M main
git push -u origin main
```

 The final section is '...or import code from another repository', which states 'You can initialize this repository with code from a Subversion, Mercurial, or TFS project.' and has an 'Import code' button. At the bottom, there is a 'ProTip!' that says 'Use the URL for this page when adding GitHub as a remote.'

renana / github-102 Public

Unwatch 1 Star 0 Fork 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

**Quick setup — if you've done this kind of thing before**

Set up in Desktop or HTTPS SSH https://github.com/renana/github-102.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# github-102" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/renana/github-102.git
git push -u origin main
```

**...or push an existing repository from the command line**

```
git remote add origin https://github.com/renana/github-102.git
git branch -M main
git push -u origin main
```

**...or import code from another repository**

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

**ProTip!** Use the URL for this page when adding GitHub as a remote.

# Sync your local and remote repositories

Let's push our local repository to the newly created remote repository:

```
> git remote add origin https://github.com/renana/github-102.git

> git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 240 bytes | 240.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/renana/github-102.git
* [new branch]    main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```



# Remote & local branches

'main' branch is now sync with the remote 'main' branch:

```
> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

# Remote & local branches

```
> git checkout my-first-branch
Switched to branch 'my-first-branch'

> git push -u origin my-first-branch
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 280 bytes | 280.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'my-first-branch' on GitHub by visiting:
remote:   https://github.com/renana/github-102/pull/new/my-first-branch
remote:
To https://github.com/renana/github-102.git
 * [new branch]    my-first-branch -> my-first-branch
Branch 'my-first-branch' set up to track remote branch 'my-first-branch' from 'origin'.
```

# Remote branches

The screenshot shows the GitHub interface for a repository named 'github-102' by user 'renana'. The repository is public. The main navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. On the right, there are buttons for Unwatch (1), Star (0), and Fork (0). The repository name 'renana / github-102' is displayed with a 'Public' badge. Below the navigation bar, a yellow banner indicates that the 'my-first-branch' has recent pushes 2 minutes ago, with a 'Compare & pull request' button. Below this, a section shows the current branch 'main' and indicates '2 branches' and '0 tags'. A commit by 'renana' is shown with the message 'creating first branch', commit hash 'a700914', and timestamp '6 hours ago', with '1 commit' indicated. Below the commit, a file 'hello-world.sh' is listed with the message 'creating first branch' and timestamp '6 hours ago'. A blue banner at the bottom of the repository section encourages adding a README. On the right side, the 'About' section is empty, and the 'Releases' and 'Packages' sections indicate no releases or packages published. The 'Languages' section shows 'Shell' at 100.0%. The footer includes copyright information for GitHub, Inc. and various links like Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

DARE TO...



# Congratulations!

You just created your second remote branch.



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)



# Local and Remote Branches

## Branches

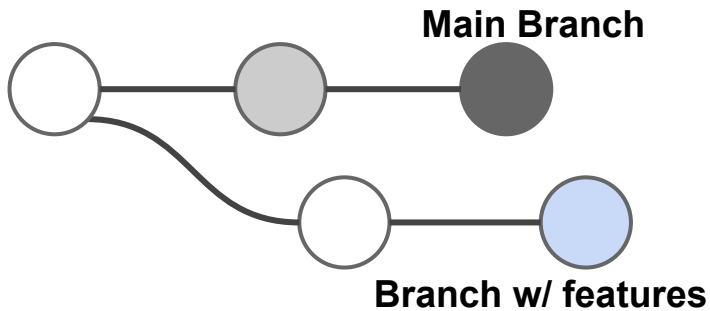
Branches allow you to test different approaches and fix different issues in the same repository.

## Local

Branch that only you (the local user) can see. It exists only on your local machine.

## Remote

Branch on the remote server. Other users can also see it.



# Contributing to open source projects

# Ways to contribute

## Creating issues

- Report a bug
- Propose a new feature

## Fixing issues

- Claim issues and submit pull requests with your work.

## Review proposed solutions

- Participate in issues discussions
- Comment on pull requests

# GitHub Issues

Why do we need them?

- Keep track of tasks and bugs
- Keep track of enhancement proposals





# Reporting an issue

Step 1: RTM – Read the manual, or more precise the contribution guide.

For example this is from

<https://github.com/github/docs/blob/main/CONTRIBUTING.md>

## Don't see your issue? Open one

If you spot something new, open an issue using a [template](#). We'll use the issue to have a conversation about the problem you want to fix.

**Note:** We cannot accept contributions to the [REST API reference documentation](#). If you spot an inaccuracy in the REST API reference documentation, open an issue in the [github/rest-api-description](#) repository.

# Reporting an issue

Step 2: follow the instructions and create an issue.

Let's create one in our own repository to get the feel of it.

# Reporting an issue

The screenshot shows the GitHub interface for creating a new issue in the repository 'renana/github-102'. The 'Issues' tab is selected in the top navigation bar. The issue title is 'My first issue'. The 'Write' tab is active, showing a text area with the content 'Learning the basics of creating issues.' and a rich text editor toolbar. A green 'Submit new issue' button is at the bottom right of the form. To the right of the form, there are sections for 'Assignees' (showing 'renana'), 'Labels' (with a 'good first issue' label selected), 'Projects' (None yet), 'Milestone' (No milestone), 'Linked pull requests' (None yet), and 'Helpful resources' (link to GitHub Community Guidelines). Two blue arrows point to the gear icons next to the 'Assignees' and 'Labels' sections.

renana / github-102 Public

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

My first issue

Write Preview

H B I

Learning the basics of creating issues.

Attach files by dragging & dropping, selecting or pasting them.

Styling with Markdown is supported

Submit new issue

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Assignees

renana

Labels

good first issue

Projects

None yet

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

Helpful resources

[GitHub Community Guidelines](#)

DARE TO...

# Reporting an issue

The screenshot shows a GitHub repository page for 'renana / github-102' (Public). The navigation bar includes links for Code, Issues (1), Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The issue title is 'My first issue #1', marked as 'Open', with a note that 'renana opened this issue now · 0 comments'. The issue body contains a comment from 'renana' (Owner) stating 'Learning the basics of creating issues.' Below this, a log shows 'renana added the good first issue label now' and 'renana self-assigned this now'. A comment input area is at the bottom with a 'Write' tab, a rich text editor, and buttons for 'Close issue' and 'Comment'. The right sidebar displays metadata: Assignees (renana), Labels (good first issue), Projects (None yet), Milestone (No milestone), Linked pull requests (None yet), and Notifications (Unsubscribe).

renana / github-102 Public

Unwatch 1 Star 0 Fork 0

<> Code Issues 1 Pull requests Actions Projects Wiki Security Insights Settings

## My first issue #1

Open renana opened this issue now · 0 comments

renana commented now Owner

Learning the basics of creating issues.

renana added the **good first issue** label now

renana self-assigned this now

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close issue Comment

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Assignees: renana

Labels: **good first issue**

Projects: None yet

Milestone: No milestone

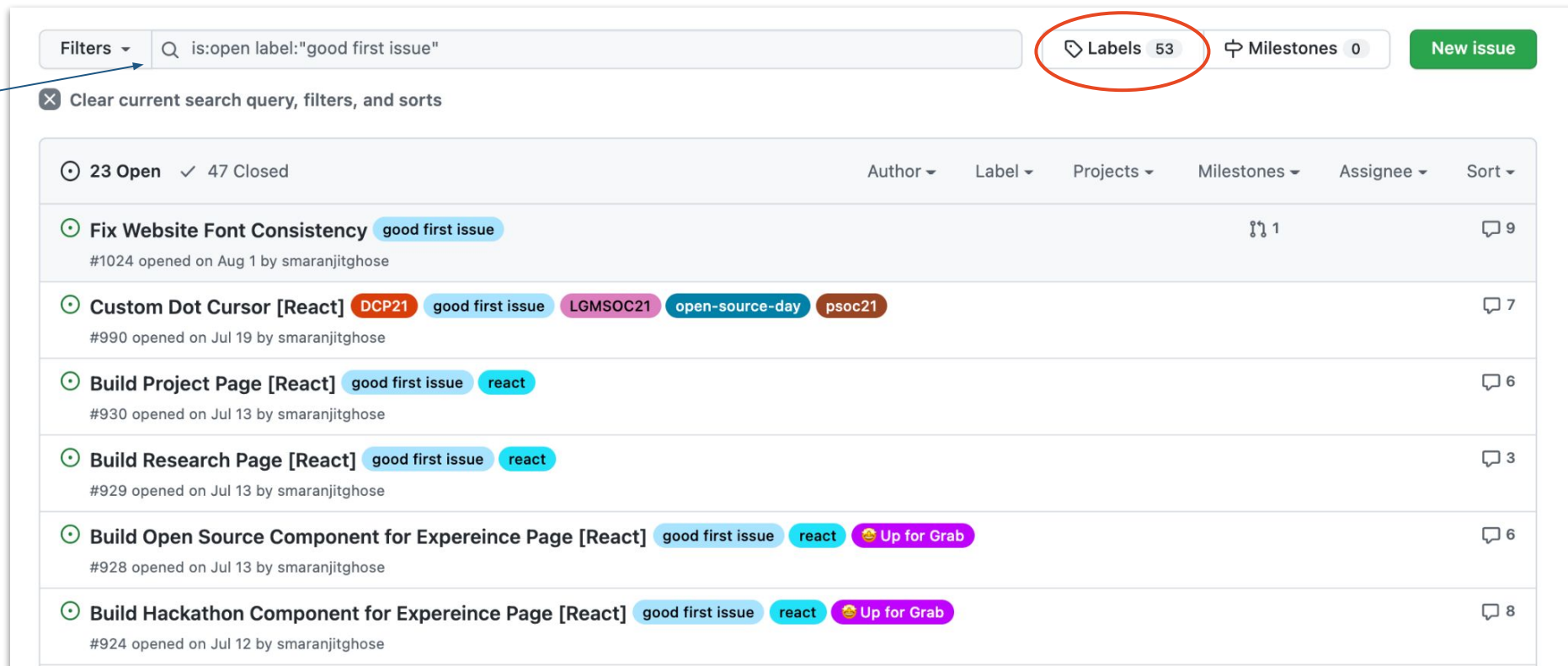
Linked pull requests: Successfully merging a pull request may close this issue. None yet

Notifications: Unsubscribe

You're receiving notifications because you're watching this repository.

# Finding an issue

Use 'good first issue' label to find an issue to work on.



The screenshot shows a GitHub search interface. The search bar contains the query "is:open label:\"good first issue\"". A red circle highlights the "Labels 53" button. A blue arrow points to the "Filters" dropdown. Below the search bar, there is a "Clear current search query, filters, and sorts" button. The results table shows a list of issues, all marked as "good first issue".

Issue Title	Labels	Comments
Fix Website Font Consistency	good first issue	9
Custom Dot Cursor [React]	DCP21, good first issue, LGMSOC21, open-source-day, psoc21	7
Build Project Page [React]	good first issue, react	6
Build Research Page [React]	good first issue, react	3
Build Open Source Component for Experience Page [React]	good first issue, react, Up for Grab	6
Build Hackathon Component for Experience Page [React]	good first issue, react, Up for Grab	8

# Claiming an issue 💪

github / docs Public

Watch 1.1k Star 6k Fork 25.3k

< > Code Issues 70 Pull requests 30 Discussions Actions Projects 2 Security Insights

## Use operationId for #anchor tags of REST API endpoint titles #139

New issue

Open 1 task done gr2m opened this issue on Jul 26, 2020 · 11 comments

gr2m commented on Jul 26, 2020 Task lists! Give feedback Contributor

**What feature or product is affected?**

☒ REST API

**What is the new or expected behavior?**

Example

<https://docs.github.com/en/rest/reference/repos#check-if-vulnerability-alerts-are-enabled-for-a-repository>

would become

<https://docs.github.com/en/rest/reference/repos#check-vulnerability-alerts>

See OpenAPI spec at

<https://github.com/github/openapi/blob/347c7c385184352692d4118b7e8a266ccf9b7db7/definitions/operations/repos/check-vulnerability-alerts.yml#L6>

**How is the old or inaccurate behavior currently documented?**

n/a

**Who does this affect?**

Every change to an #anchor name for endpoint titles requires a change in the code, due to the error handlers which include a URL to the documentation. The endpoint titles are more prone to changes, while operation IDs should be changed much less

**Assignees**

No one assigned

**Labels**

engineering good first issue help wanted

**Projects**

Docs team reviews  
Help wanted

**Milestone**

No milestone

**Linked pull requests**

Successfully merging a pull request may close this issue.

None yet

**Notifications** Customize

Subscribe

DARE TO...



# Claiming an issue

## Commit pull request

---


- Please choose an interested issue, or create a new issue and then settle a correct label.
- Reply a deadline message to pickup this issue.
- Find a mentor in [Core developers list](#), he will give you feedback for design and implements.
- Fork to your github repo and begin to work.
- Please follow Sharding's [Development conventions](#), and complete check before pull request submit.
- Submit a pull request to master branch when finished.
- Mentor will do code review and discuss some details, include design, implement, performance and code style. Code will be merged until mentor accepted.
- Finally, congratulations that you have become the official contributor for Sharding!

# Claiming an issue

## Build Hackathon Component for Experience Page [React] #924

New issue

Open smaranjitghose opened this issue on Jul 12 · 8 comments




smaranjitghose commented on Jul 12


Owner

Reference

- The title of the section should be **Hackathon Participation**
- The UI should be more or less the same as in the master branch of the project (Feel free to modify the cards or anything else if you wish)
- Do not change the background color of the page ( in the reference it is a shade of orange)
- The Color of the scrollbar of the timeline should be the same as that of the headings of the page
- Do not put Lorem Ipsum and random Images. Use the data given for the same in the master branch
- The data should be dynamically inserted using a variable as done through experience.js in the current version.
- The title should be the name of the Hackathon
- Add a subtitle for the role (Judge/Mentor/Speaker/Winner/Runnerup)
- Images/illustrations should be placed inside assets/images/hackathons/




smaranjitghose added good first issue react open-source-day DCP21 psoc21 Up for Grab labels on Jul 12



souravseal99 commented on Jul 13

Contributor


I would like to work on this.



deeqakkk commented on Jul 13

Collaborator

@souravseal99 assigned for 3 days.

 1

Assignees

No one assigned

Labels

good first issue react Up for Grab

Projects

None yet

Milestone


No milestone

Linked pull requests

Successfully merging a pull request may close this issue.





None yet

Notifications Customize

 Subscribe

You're not receiving notifications from this thread.

4 participants

# Breakout rooms tasks

Round 1 of 2 – 10 minutes

1. Clone the repository assigned to your room
2. Find an issue and claim it
3. Make sure no-one else claimed it first
4. Work on the fix and create a pull request (PR)
5. Ask your peers to review and approve it
6. Merge your PR
7. Check if your peers need help

# Survey time

- How much were you able to finish in this time?
  - I wasn't able to do anything
  - Claimed an issue
  - Claimed an issue and started working on it
  - Claimed an issue and opened a PR with solution
  - Claimed an issues, merged the solution and closed the issue
  - Completed several issues

# Reviewing a PR

- Comments should be relevant and discussion professional
- Comments should be clear and include all necessary information
- If you are not sure, tag a person that knows and can confirm your suspicion
- **Remember the code-of-conduct!**

# Adding comments to a PR

Add a comment by clicking the '+':

The screenshot shows a GitHub Pull Request (PR) titled "Changed message to Hello World1 #2". The PR is open, showing a diff for the file "hello-world.sh". The diff shows a change from "echo Hello World" to "echo Hello World1". A blue arrow points to the "+" icon next to the new line of code, indicating where to click to add a comment. The interface includes navigation tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The PR details section shows the commit history, file changes, and a "Review changes" button. The footer contains copyright information and links to GitHub's terms, privacy, security, status, docs, contact, pricing, API, training, blog, and about pages.

renana / github-102 Public

Unwatch 1 Star 0 Fork 0

<> Code Issues 1 Pull requests 1 Actions Projects Wiki Security Insights Settings

Changed message to Hello World1 #2

Edit <> Code

Open renana wants to merge 1 commit into main from my-first-branch

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -1

Changes from all commits File filter Conversations Jump to 0 / 1 files viewed Review changes

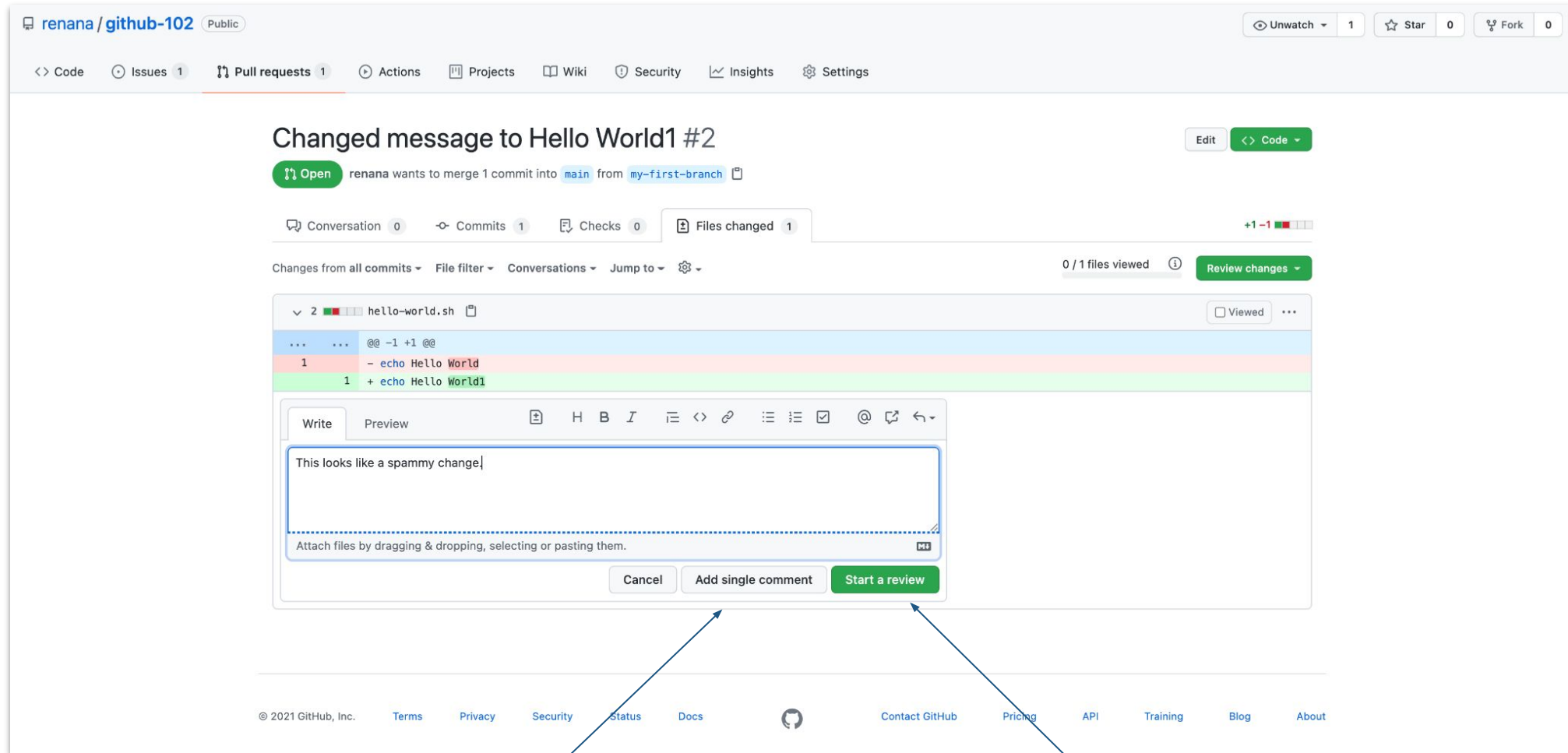
2 hello-world.sh

```
... @@ -1 +1 @@  
- echo Hello World  
+ echo Hello World1
```

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About



# Adding comments to a PR



The screenshot displays a GitHub Pull Request (PR) for the repository `renana/github-102`. The PR title is "Changed message to Hello World1 #2". The interface shows the PR details, including the commit being merged from `my-first-branch` into `main`. Below the PR details, the file `hello-world.sh` is shown with a diff view. The diff shows a change from `echo Hello World` to `echo Hello World1`. A comment box is open, showing the text "This looks like a spammy change." and a "Write" tab. The comment box has a "Preview" tab and a "Write" tab. Below the comment box, there are buttons for "Cancel", "Add single comment", and "Start a review". Two blue arrows point from the bottom of the image to the "Add single comment" and "Start a review" buttons.

renana / github-102 Public

Unwatch 1 Star 0 Fork 0

<> Code Issues 1 Pull requests 1 Actions Projects Wiki Security Insights Settings

Changed message to Hello World1 #2 Edit <> Code

Open renana wants to merge 1 commit into main from my-first-branch

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -1

Changes from all commits File filter Conversations Jump to 0 / 1 files viewed Review changes

2 hello-world.sh Viewed

```
... @@ -1,1 +1,1 @@
1 - echo Hello World
1 + echo Hello World1
```

Write Preview H B I <> @

This looks like a spammy change.

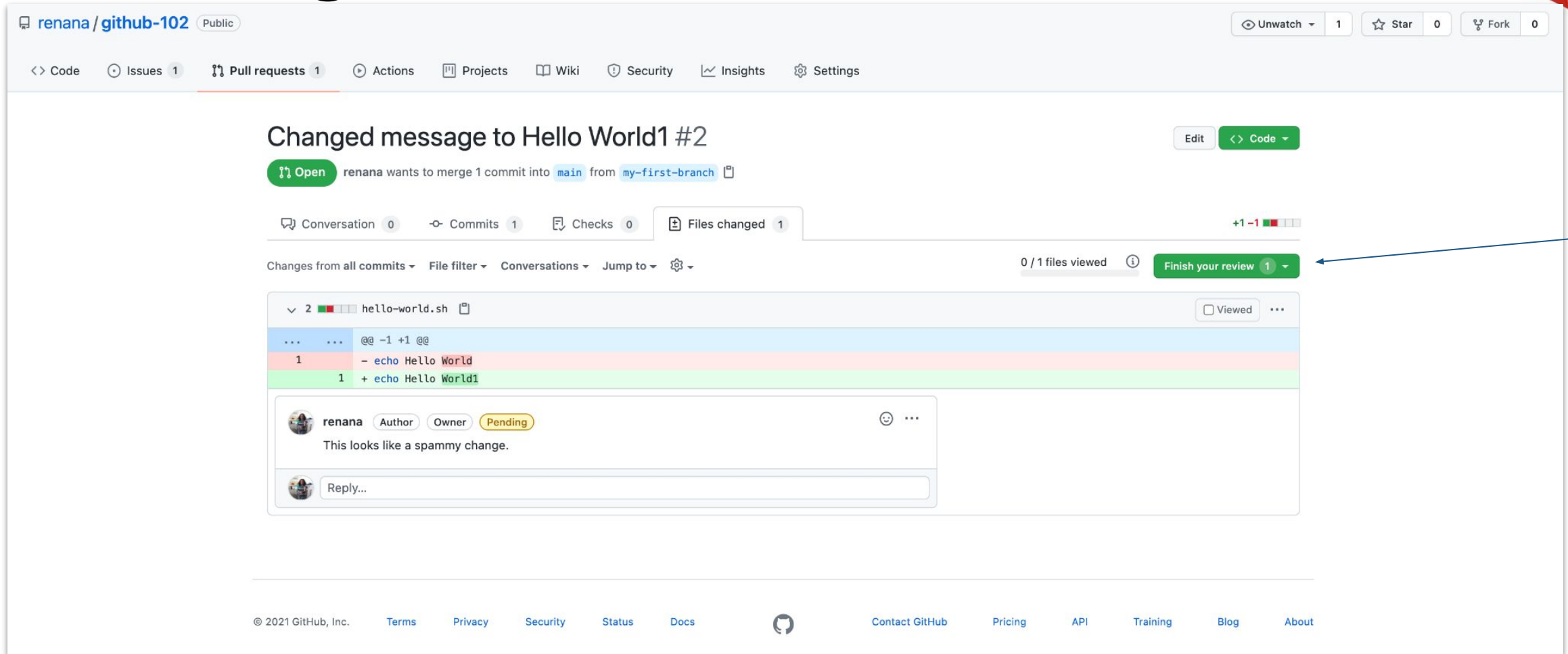
Attach files by dragging & dropping, selecting or pasting them.

Cancel Add single comment Start a review

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

**DARE** TO...

# Adding comments to a PR



The screenshot shows a GitHub Pull Request (PR) titled "Changed message to Hello World1 #2" by user "renana". The PR is open and shows a diff for the file "hello-world.sh". The diff shows a change from "echo Hello World" to "echo Hello World1". A review comment by "renana" is visible, stating "This looks like a spammy change." The comment is marked as "Pending". The PR page includes navigation tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The PR details section shows the conversation, commits, checks, and files changed. The file "hello-world.sh" is expanded, showing the diff. The review comment section is at the bottom of the diff view.

renana / github-102 Public

Unwatch 1 Star 0 Fork 0

<> Code Issues 1 Pull requests 1 Actions Projects Wiki Security Insights Settings

## Changed message to Hello World1 #2

Edit <> Code

Open renana wants to merge 1 commit into main from my-first-branch

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -1

Changes from all commits File filter Conversations Jump to 0 / 1 files viewed Finish your review 1

2 hello-world.sh Viewed

```
... @@ -1,1 @@
1 - echo Hello World
1 + echo Hello World1
```

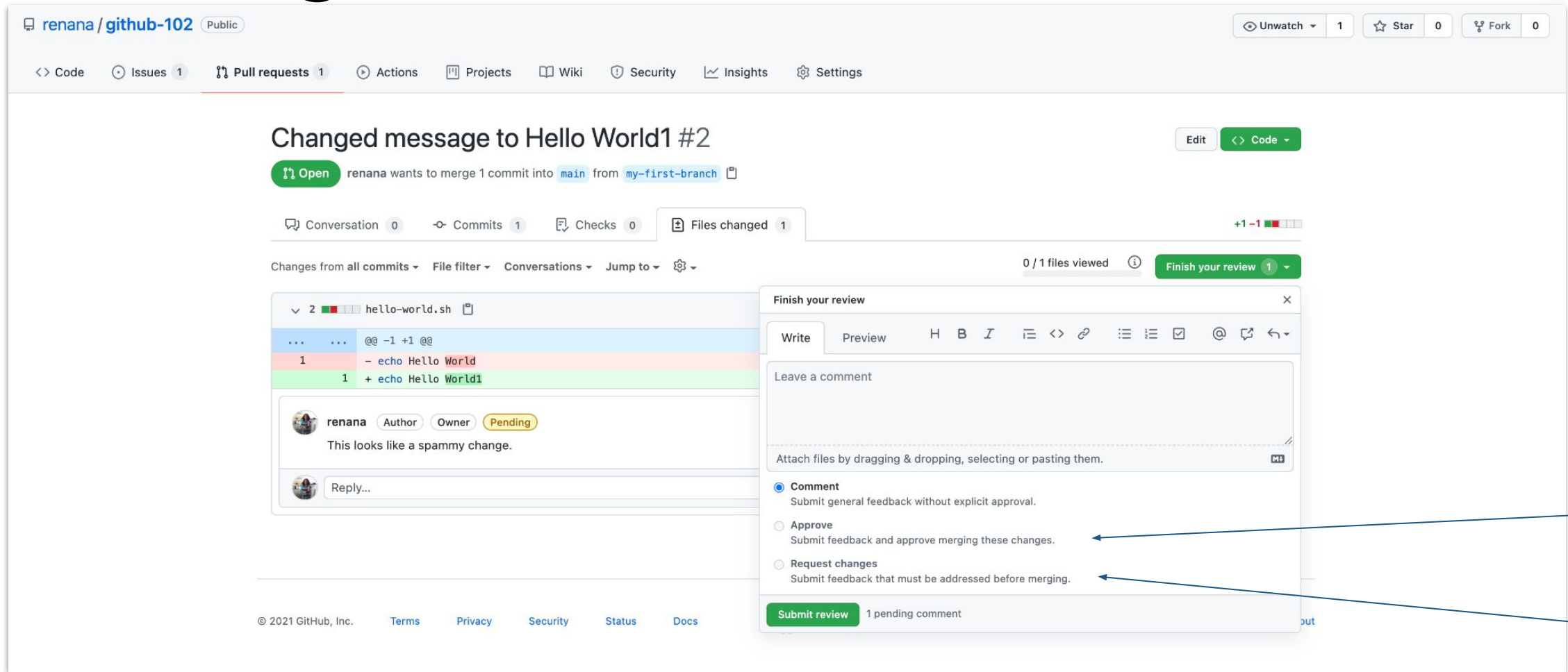
renana Author Owner Pending

This looks like a spammy change.

Reply...

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

# Adding comments to a PR



The screenshot displays a GitHub Pull Request (PR) for the repository `renana / github-102`. The PR title is "Changed message to Hello World1 #2". The PR is open, showing a change from `main` to `my-first-branch`. The change is in the file `hello-world.sh`, where a line was added: `+ echo Hello World1`. The PR is pending review, with a comment from the author: "This looks like a spammy change." A review modal is open, showing options to "Comment", "Approve", or "Request changes". The "Comment" option is selected, and the text "Leave a comment" is visible. The modal also shows a "Submit review" button and a "1 pending comment" indicator. Arrows point from the "Approve" and "Request changes" options to the right side of the image.

renana / github-102 Public

<> Code Issues 1 Pull requests 1 Actions Projects Wiki Security Insights Settings

Changed message to Hello World1 #2

Open renana wants to merge 1 commit into main from my-first-branch

Conversation 0 Commits 1 Checks 0 Files changed 1

Changes from all commits File filter Conversations Jump to 0 / 1 files viewed Finish your review 1

2 hello-world.sh

```
... @@ -1,1 @@
1 - echo Hello World
1 + echo Hello World1
```

renana Author Owner Pending

This looks like a spammy change.

Reply...

Finish your review

Write Preview H B I

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

☒ **Comment**  
Submit general feedback without explicit approval.

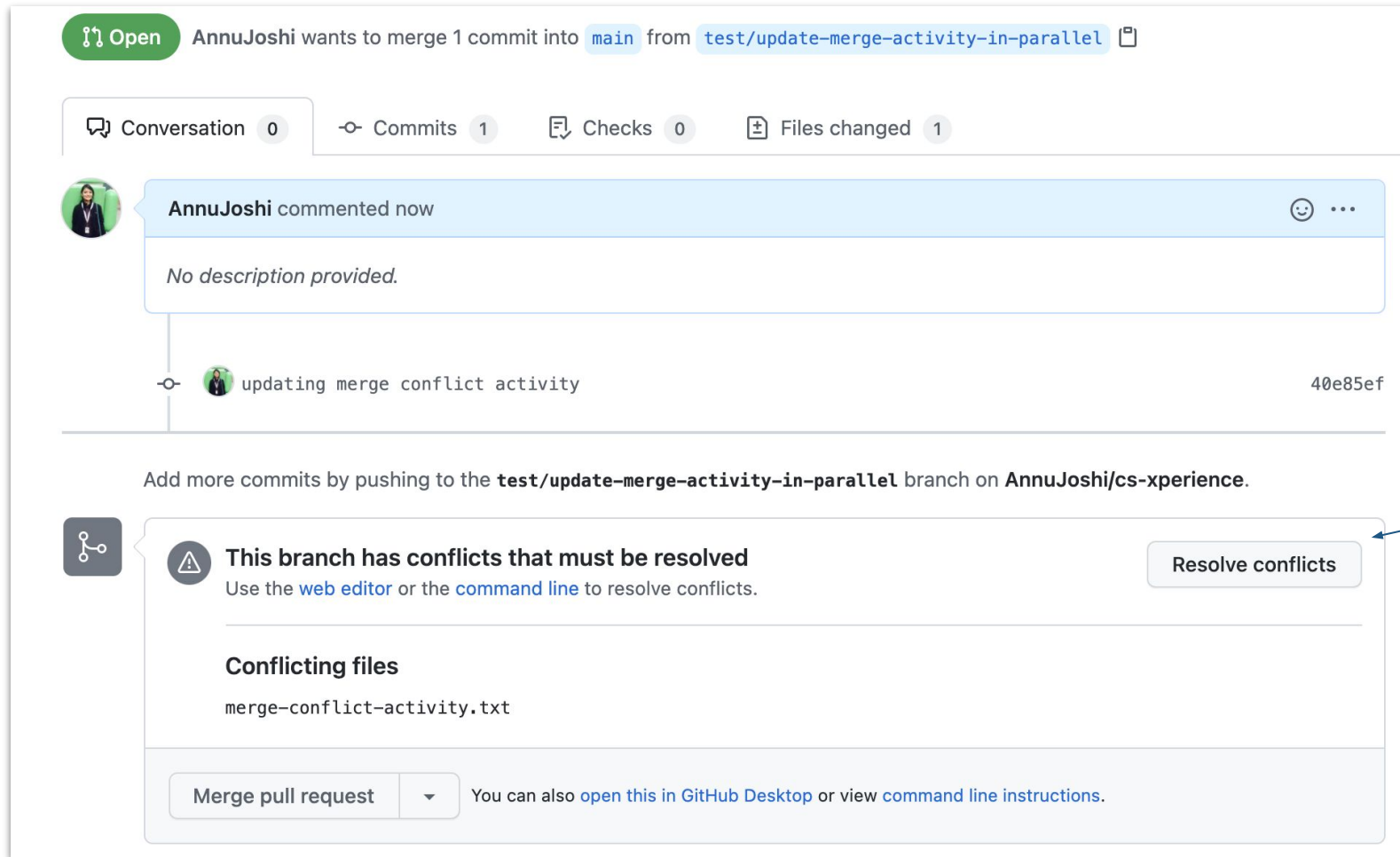
☐ **Approve**  
Submit feedback and approve merging these changes.

☐ **Request changes**  
Submit feedback that must be addressed before merging.

Submit review 1 pending comment

DARE TO...



# Resolving a merge conflict



The screenshot shows a GitHub pull request interface. At the top, a green button labeled 'Open' is next to the text 'AnnuJoshi wants to merge 1 commit into `main` from `test/update-merge-activity-in-parallel`'. Below this, a navigation bar shows 'Conversation 0', 'Commits 1', 'Checks 0', and 'Files changed 1'. A comment from 'AnnuJoshi' is visible, stating 'No description provided.' Below the comment, a commit is shown as 'updating merge conflict activity' with a hash of '40e85ef'. A message below the commit states: 'Add more commits by pushing to the `test/update-merge-activity-in-parallel` branch on `AnnuJoshi/cs-xperience`.' A large warning box with a red triangle icon contains the text: 'This branch has conflicts that must be resolved. Use the [web editor](#) or the [command line](#) to resolve conflicts.' A 'Resolve conflicts' button is located to the right of this warning. Below the warning, under the heading 'Conflicting files', the file 'merge-conflict-activity.txt' is listed. At the bottom, there is a 'Merge pull request' button and a note: 'You can also [open this in GitHub Desktop](#) or view [command line instructions](#).'

# Resolving a merge conflict



Resolving conflicts between `activity/merge-c...` and `main` and committing changes → `activity/merge-c...`

1 conflicting file	<code>merge-conflict-activity.txt</code> 1 conflict Prev ^ Next v ⚙ Mark as resolved
 <code>merge-conflict-... merge-conflict- activity.txt</code>	<pre>1 &lt;&lt;&lt;&lt;&lt;&lt;&lt; activity/merge-conflict 2 Hello there! We are learning about merge conflict and many more things today! 3 I am so excited, we have a plan! Woohooo! 4 ===== 5 Hello there! We are going to learn about merge conflicts at OSD celebrations today! 6 I am so excited! 7 &gt;&gt;&gt;&gt;&gt;&gt;&gt; main 8</pre> 

# Resolving a merge conflict

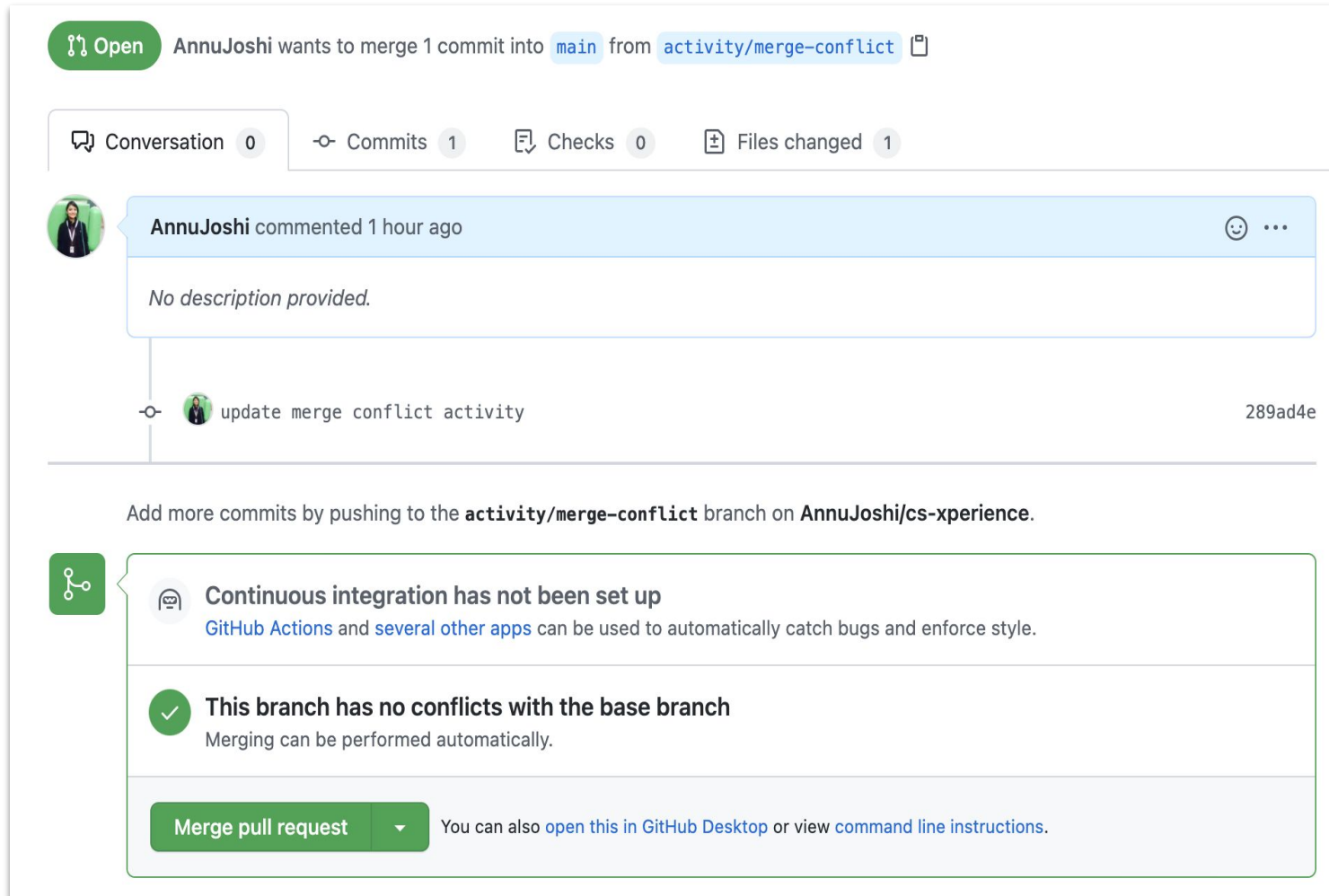
Resolving conflicts between `activity/merge-c...` and `main` and committing changes → `activity/merge-c...` Commit merge



This merge commit will be associated with annuhansa94@gmail.com.





1 conflicting file	merge-conflict-activity.txt	✓ Resolved
merge-conflict-activit... merge-conflict-activity.txt	1 Hello there! We are learning about merge conflict and many more things today! 2 I am so excited, we have a plan! Woohooo! 3 4	 






# Resolving a merge conflict





 Open AnnuJoshi wants to merge 1 commit into `main` from `activity/merge-conflict` 


 Conversation 0  Commits 1  Checks 0  Files changed 1


 AnnuJoshi commented 1 hour ago  


No description provided.



  update merge conflict activity 289ad4e

Add more commits by pushing to the `activity/merge-conflict` branch on `AnnuJoshi/cs-xperience`.



 **Continuous integration has not been set up**  
[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

 **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](#).

# Breakout rooms tasks

Round 2 of 2

1. Clone the repository assigned to your room
2. Find an issue and claim it
3. Make sure no-one else claimed it first
4. Work on the fix and create a pull request (PR)
5. Ask your peers to review and approve it
6. Merge your PR
7. Check if your peers need help
8. Assign a person to notify me that your room is ready

# Survey time

- How much were you able to finish in this time?
  - I wasn't able to do anything
  - Claimed an issue
  - Claimed an issue and started working on it
  - Claimed an issue and opened a PR with solution
  - Claimed an issues, merged the solution and closed the issue
  - Completed several issues

# Good open source citizenship

# Working as a team

- Were all PRs reviewed?
- Is anyone blocked?
- Did you encounter merge conflicts?



# Dos when contributing to open source projects

- Contribute to open source projects
- Read the project's code-of-conduct before starting to contribute to it
- Report any violation of the code-of-conduct
- Follow the contributing guide



# Dos when contributing to open source projects

- Open issues to report bugs and suggest improvements, even if you will not be able to work on them
- Before opening an issue, ensure there isn't an existing one covering this already
- Review other people's pull requests
- Claim issues before starting to work on them

# Dos when contributing to open source projects

- Address comments and continue discussions in issues and pull requests
- Report progress on claimed issues, especially when blocked
- Connect with other contributors

# Don'ts when contributing to open source projects

- Don't violate the code-of-conduct
- Don't forget to update progress on claimed issues
- Personal preference issues, pull requests or comments are unprofessional and should rarely be used

# Thank You

**Annu Joshi**



@nibblepuff



@annujoshi

**Renana Yacobi**



@RenanaYacobi



@renanayacobi