

Communicating on GitHub

You can discuss specific projects and changes, as well as broader ideas or team goals, using different types of discussions on GitHub.

In this article

[Introduction](#)

[GitHub Issues](#)

[Pull requests](#)

[GitHub Discussions](#)

[Team discussions](#)

[Which discussion tool should I use?](#)

[Scenarios for issues](#)

[Scenarios for pull requests](#)

[Scenarios for GitHub Discussions](#)

[Scenarios for team discussions](#)

[Next steps](#)

Introduction

GitHub provides built-in collaborative communication tools allowing you to interact closely with your community. This quickstart guide will show you how to pick the right tool for your needs.



You can create and participate in issues, pull requests, GitHub Discussions, and team discussions, depending on the type of conversation you'd like to have.

GitHub Issues

- are useful for discussing specific details of a project such as bug reports, planned improvements and feedback.
- are specific to a repository, and usually have a clear owner.
- are often referred to as GitHub's bug-tracking system.

Pull requests

- allow you to propose specific changes.
- allow you to comment directly on proposed changes suggested by others.
- are specific to a repository.

GitHub Discussions

- are like a forum, and are best used for open-form ideas and discussions where collaboration is important.
- may span many repositories.
- provide a collaborative experience outside the codebase, allowing the brainstorming of ideas, and the creation of a community knowledge base.
- often don't have a clear owner.
- often do not result in an actionable task.

Team discussions



- can be started on your team's page for conversations that span across projects and don't belong in a specific issue or pull request. Instead of opening an issue in a repository to discuss an idea, you can include the entire team by having a conversation in a team discussion.
- allow you to hold discussions with your team about planning, analysis, design, user research and general project decision making in one place.

Which discussion tool should I use?

Scenarios for issues

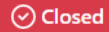
- I want to keep track of tasks, enhancements and bugs.
- I want to file a bug report.
- I want to share feedback about a specific feature.
- I want to ask a question about files in the repository.

Issue example

This example illustrates how a GitHub user created an issue in our documentation open source repository to make us aware of a bug, and discuss a fix.



Blue link text in notices is unreadable due to blue background #4703

[Edit](#)[New issue](#)

Closed

contributocat opened this issue 20 days ago · 16 comments



contributocat commented 20 days ago

Member



What is the current behavior?



What changes are you suggesting?

Use a more visible link color in notices.



1



2

Assignees



No one—assign yourself

Labels



design

engineering

localization

Projects



Docs team reviews

Done ▾

Docs Engineering - Task Bo...

Done ▾

Tables



None yet

Milestone



No milestone

- A user noticed that the blue color of the banner at the top of the page in the Chinese version of the GitHub Docs makes the text in the banner unreadable.
- The user created an issue in the repository, stating the problem and suggesting a fix (which is, use a different background color for the banner).
- A discussion ensues, and eventually, a consensus will be reached about the fix to apply.
- A contributor can then create a pull request with the fix.



Scenarios for pull requests

- I want to fix a typo in a repository.
- I want to make changes to a repository.
- I want to make changes to fix an issue.
- I want to comment on changes suggested by others.

Pull request example


This example illustrates how a GitHub user created a pull request in our documentation open source repository to fix a typo.

In the **Conversation** tab of the pull request, the author explains why they created the pull request.



Fix typo #6891

[Edit](#)[Open with ▾](#)

 **Merged** octocat merged 2 commits into `github:main` from `contributocat:patch-1` 18 days ago

[Conversation 4](#)[Commits 2](#)[Checks 31](#)[Files changed 1](#)+1 -1

contributocat commented 18 days ago

[Contributor](#)

Why:

I saw a small typo in the documentation.

What's being changed:

Just fixing a single typo.

Check off the following:

- ☐ I have reviewed my changes in staging. (look for the **deploy-to-heroku** link in your pull request, then click **View deployment**)
- ☒ For content changes, I have reviewed the [localization checklist](#)
- ☒ For content changes, I have reviewed the [Content style guide for GitHub Docs](#).

Reviewers



octocat



Assignees



No one—assign yourself

Labels



ready to merge

Projects



Docs team reviews

Done ▾

Tables



None yet



Fix typo

Verified

✓ 8f3ca61

The **Files changed** tab of the pull request shows the implemented fix.

Fix typo #6891

Edit

Open with ▼

Merged octocat merged 2 commits into `github:main` from `contributocat:patch-1` 18 days ago

Conversation 4

Commits 2

Checks 31

Files changed 1

+1 -1 ■ ■ ■ ■

File filter ▾ Jump to ▾ ⚙ ▾

0 / 1 files viewed

Review changes ▼

content/actions/reference/context-and-expression-syntax-for-github-actions.md

Viewed

@@ -323,7 +323,7 @@ Returns `true` if `searchString` ends with `searchValue`. This function is not c

323

323

324

324

325

325

326

326

327

327

```
`format( string, replaceValue0, replaceValue1, ..., replaceValueN)`
```

```
- Replaces values in the `string`, with the variable `replaceValueN`. Variables in the `string` are specified using the `{N}` syntax, where `N` is an integer. You must specify at least one `replaceValue` and `string`. There is no maximum for the number of variables (`replaceValueN`) you can use. Escape curly braces using double braces.
```

```
`format( string, replaceValue0, replaceValue1, ..., replaceValueN)`
```

```
+ Replaces values in the `string`, with the variable `replaceValueN`. Variables in the `string` are specified using the `{N}` syntax, where `N` is an integer. You must specify at least one `replaceValue` and `string`. There is no maximum for the number of variables (`replaceValueN`) you can use. Escape curly braces using double braces.
```

- This contributor notices a typo in the repository.
- The user creates a pull request with the fix.
- A repository maintainer reviews the pull request, comments on it, and merges it.

Scenarios for GitHub Discussions

- I have a question that's not necessarily related to specific files in the repository.
- I want to share news with my collaborators, or my team.
- I want to start or participate in an open-ended conversation.
- I want to make an announcement to my community.



GitHub Discussions example

This example shows the GitHub Discussions welcome post for the GitHub Docs open source repository, and illustrates how the team wants to collaborate with their community.

↑
8

Welcome to GitHub Docs Discussions! #158

octocat started this conversation in General

Edit

 **octocat** on 6 Oct 2020 Maintainer

 **Welcome to GitHub Docs Discussions!**

We're using Discussions as a place to connect with other members of our community. We hope that you:

- Ask questions you're wondering about.
- Share ideas.
- Get help with a PR you're working on.
- Engage with other community members.
- Build the docs community together! 🙌

To get started, comment below with an introduction of yourself and tell us about what you do with GitHub Docs!



 26

 1

 8

 34

 4

Category

 General

27 participants



and others

Notifications

 **Subscribe**

You're not receiving notifications from this thread.

 Lock conversation

 Transfer discussion

 Edit pinned discussion

 Unpin discussion

 Delete discussion

Beta Give feedback

24 comments · 35 replies

Oldest

Newest

Top

↑
1

 **Contributocat** on 7 Oct 2020

My name is Contributocat and I haven't yet gotten started using GitHub Docs but I'll see where I can jump in.



 3

 7

4 replies

This community maintainer started a discussion to welcome the community, and to ask members to introduce themselves. This post fosters an inviting atmosphere for visitors and contributors. The post also clarifies that the team's happy to help with contributions to the repository.

<https://docs.github.com/en/get-started/quickstart/communicating-on-github>

8/11

Scenarios for team discussions

- I have a question that's not necessarily related to specific files in the repository.
- I want to share news with my collaborators, or my team.
- I want to start or participate in an open-ended conversation.
- I want to make an announcement to my team.

As you can see, team discussions are very similar to GitHub Discussions. For GitHub.com, we recommend using GitHub Discussions as the starting point for conversations. You can use GitHub Discussions to collaborate with any community on GitHub. If you are part of an organization, and would like to initiate conversations within your organization or team within that organization, you should use team discussions.

Team discussion example

This example shows a team post for the `octo-team` team.





octo-team

@github/octo-repo

28 members



26 are members of child teams

Unwatch

Member statuses



Load more...

Teams

9



Create a team post for @github/octo-repo

Recent

Pinned 3

Docs Update for May 3, 2021



octocat

19 hours ago • edited

How can we add delight to our work life and carve out our "home" together? Mona started adding games to our calendars, and we can all follow her lead and do the same. We can also eat ice cream straight out of the carton together. Maybe while playing games?



Docs happenings!

What a great example of collaboration between our two subteams in [this blog post](#) about how we use GitHub Actions in the docs!

We had our April Docs All Hands on Friday. You can check out [the recording](#) and [slides](#).

The `octocat` team member posted a team discussion, informing the team of various things:

- A team member called Mona started remote game events.
- There is a blog post describing how the teams use GitHub Actions to produce their docs.
- Material about the April All Hands is now available for all team members to view.

Next steps



These examples showed you how to decide which is the best tool for your conversations on GitHub. But this is only the beginning; there is so much more you can do to tailor these tools to your needs.

For issues, for example, you can tag issues with labels for quicker searching and create issue templates to help contributors open meaningful issues. For more information, see "[About issues](#)" and "[About issue and pull request templates](#)."

For pull requests, you can create draft pull requests if your proposed changes are still a work in progress. Draft pull requests cannot be merged until they're marked as ready for review. For more information, see "[About pull requests](#)."

For GitHub Discussions, you can set up a code of conduct and pin discussions that contain important information for your community. For more information, see "[About discussions](#)."

For team discussions, you can edit or delete discussions on a team's page, and you can configure notifications for team discussions. For more information, see "[About team discussions](#)."

