Searching discussions

In this article

You can search for discussions on GitHub Enterprise Server and narrow the results using search qualifiers.

About searching for discussions 🔗

You can search for discussions globally across all of GitHub Enterprise Server, or search for discussions within a particular organization or repository. For more information, see "About searching on GitHub."

> **Tips:**
>
> - This article contains links to example searches on the GitHub.com website, but you can use the same search filters with GitHub Enterprise Server. In the linked example searches, replace `github.com` with the hostname for your GitHub Enterprise Server instance.
> - For a list of search syntaxes that you can add to any search qualifier to further improve your results, see "Understanding the search syntax".
> - Use quotations around multi-word search terms. For example, if you want to search for issues with the label "In progress," you'd search for `label:"in progress"` . Search is not case sensitive.

Search by the title, body, or comments 🔗

With the `in` qualifier you can restrict your search for discussions to the title, body, or comments. You can also combine qualifiers to search a combination of title, body, or comments. When you omit the `in` qualifier, GitHub Enterprise Server searches the title, body, and comments.

Qualifier	Example
`in:title`	**welcome in:title** matches discussions with "welcome" in the title.
`in:body`	**onboard in:title,body** matches discussions with "onboard" in the title or body.
`in:comments`	**thanks in:comments** matches discussions with "thanks" in the comments for the discussion.

Search within a user's or organization's repositories 🔗

To search discussions in all repositories owned by a certain user or organization, you can use the `user` or `org` qualifier. To search discussions in a specific repository, you can use the `repo` qualifier.

Qualifier	Example
`user:USERNAME`	**user:octocat feedback** matches discussions with the word "feedback" from repositories owned by @octocat.
`org:ORGNAME`	**org:github** matches discussions in repositories owned by the GitHub organization.
`repo:USERNAME/REPOSITORY`	**repo:nodejs/node created:<2021-01-01** matches discussions from @nodejs' Node.js runtime project that were created before January 2021.

Search by open or closed state 🔗

You can filter discussions based on whether they're open or closed using the `is` qualifier.

Qualifier	Example
`is:open`	**performance is:open is:discussion** matches open discussions with the word "performance."
`is:closed`	**android is:closed** matches closed discussions with the word "android."

Search based on whether a discussion was answered 🔗

You can search for a discussion that has been answered using the `is` qualifier.

Qualifier	Example
`is:answered`	**performance is:answered is:discussion** matches answered discussions with the word "performance."

`is:unanswered`	**android is:unanswered** matches unanswered discussions with the word "android."

Search based on whether a discussion is locked 🔗

You can search for a discussion that has been locked using the `is` qualifier. For more information, see "Moderating discussions."

Qualifier	Example
`is:locked`	**"code of conduct" is:locked is:discussion** matches discussions with the words "code of conduct" that have been locked.
`is:unlocked`	**code of conduct is:unlocked is:discussion** matches discussions with the words "code of conduct" that are unlocked.

Filter by repository visibility 🔗

You can filter by the visibility of the repository containing the discussions using the `is` qualifier. For more information, see "About repositories."

Qualifier	Example
`is:public`	**is:public** matches discussions in public repositories.
`is:private`	**is:private tiramisu** matches discussions that contain the word "tiramisu" in private repositories you can access.

Search by author 🔗

The `author` qualifier finds discussions created by a certain user.

Qualifier	Example
`author:USERNAME`	**cool author:octocat** matches discussions with the word "cool" that were created by @octocat.
`in:body` `author:USERNAME`	**bootstrap in:body author:octocat** matches discussions created by @octocat that contain the word "bootstrap" in the body.

Search by commenter 🔗

The `commenter` qualifier finds discussions that contain a comment from a certain user.

Qualifier	Example
`commenter:USERNAME`	**github commenter:becca org:github** matches discussions in repositories owned by GitHub, that contain the word "github," and have a comment by @becca.

Search by user who has answered a discussion 🔗

The `answered-by` qualifier finds discussions where a certain user's comment was marked as an answer.

Qualifier	Example
answered-by:*USERNAME*	**cool answered-by:octocat** matches discussions with the word "cool" that were answered by @octocat.

Search by a user that's involved in a discussion 🔗

You can use the `involves` qualifier to find discussions that involve a certain user. The qualifier returns discussions that were either created by a certain user, mention the user, or contain comments by the user. The `involves` qualifier is a logical OR between the `author`, `mentions`, and `commenter` qualifiers for a single user.

Qualifier	Example
involves:*USERNAME*	**involves:becca involves:octocat** matches discussions either @becca or @octocat are involved in.
in:body involves:*USERNAME*	**NOT beta in:body involves:becca** matches discussions @becca is involved in that do not contain the word "beta" in the body.

Search by number of comments 🔗

You can use the `comments` qualifier along with greater than, less than, and range qualifiers to search by the number of comments. For more information, see "[Understanding the search syntax](#)."

Qualifier	Example
comments:*n*	**comments:>100** matches discussions with more than 100 comments.
comments:*n*	**comments:500..1000** matches discussions with comments ranging from 500 to 1,000.

Search by when a discussion was created or last updated 🔗

You can filter discussions based on times of creation, or when the discussion was last updated. For discussion creation, you can use the `created` qualifier; to find out when an discussion was last updated, use the `updated` qualifier.

Both qualifiers take a date as a parameter. Date formatting must follow the [ISO8601](#) standard, which is `YYYY-MM-DD` (year-month-day). You can also add optional time information `THH:MM:SS+00:00` after the date, to search by the hour, minute, and second. That's `T`, followed by `HH:MM:SS` (hour-minutes-seconds), and a UTC offset (`+00:00`).

When you search for a date, you can use greater than, less than, and range qualifiers to further filter results. For more information, see "Understanding the search syntax."

Qualifier	Example
`created:`*`YYYY-MM-DD`*	**created:>2020-11-15** matches discussions that were created after November 15, 2020.
`updated:`*`YYYY-MM-DD`*	**weird in:body updated:>=2020-02-01** matches discussions with the word "weird" in the body that were updated after December 2020.

Search by category ∥

You can filter discussions by specific discussions categories.

Qualifier	Example
`category:`*`CATEGORYNAME`*	**category:Ideas** matches discussions categories that match the name "Ideas".

Search by label ∥

You can filter discussions by specific labels that are applied to discussions.

Qualifier	Example
`label: `*`"LABEL NAME"`*	**label:"Product Feedback"** matches discussions that match the label "Product Feedback".

Further reading ∥

- "Sorting search results"