

# Searching issues and pull requests

## In this article

- Search only issues or pull requests
- Search by the title, body, or comments
- Search within a user's or organization's repositories
- Search by open or closed state
- Search by the reason an issue was closed
- Filter by repository visibility
- Search by author
- Search by assignee
- Search by mention
- Search by team mention
- Search by commenter
- Search by a user that's involved in an issue or pull request
- Search for linked issues and pull requests
- Search by label
- Search by milestone
- Search by project board
- Search by commit status
- Search by commit SHA
- Search by branch name
- Search by language
- Search by number of comments
- Search by number of interactions
- Search by number of reactions
- Search for draft pull requests
- Search by pull request review status and reviewer
- Search by when an issue or pull request was created or last updated
- Search by when an issue or pull request was closed
- Search by when a pull request was merged
- Search based on whether a pull request is merged or unmerged
- Search based on whether a repository is archived
- Search based on whether a conversation is locked
- Search by missing metadata
- Further reading

---

You can search for issues and pull requests on GitHub Enterprise Server and narrow the results using these search qualifiers in any combination.

You can search for issues and pull requests globally across all of GitHub Enterprise Server, or search for issues and pull requests within a particular organization. For more information, see "[About searching on GitHub](#)."

### Tips:

- This article contains example searches on the GitHub.com website, but you can use the same search filters on 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.
- Use a minus (hyphen) symbol to exclude results that match a qualifier. For example, to ignore issues created by the "octocat" user, you'd use `-author:octocat` in your search.
- You can focus your cursor on the search bar above the issue or pull request list with a keyboard shortcut. For more information, see "[Keyboard shortcuts](#)."

## Search only issues or pull requests

By default, GitHub Enterprise Server search will return both issues and pull requests. However, you can restrict search results to just issues or pull requests using the `type` or `is` qualifier.

Qualifier	Example
<code>type:pr</code>	<a href="#">cat type:pr</a> matches pull requests with the word "cat."
<code>type:issue</code>	<a href="#">github commenter:defunkt type:issue</a> matches issues that contain the word "github," and have a comment by @defunkt.
<code>is:pr</code>	<a href="#">event is:pr</a> matches pull requests with the word "event."
<code>is:issue</code>	<a href="#">is:issue label:bug is:closed</a> matches closed issues with the label "bug."

## Search by the title, body, or comments

With the `in` qualifier you can restrict your search to the title, body, comments, or any combination of these. When you omit this qualifier, the title, body, and comments are all searched.

Qualifier	Example
<code>in:title</code>	<a href="#">warning in:title</a> matches issues with "warning" in their title.
<code>in:body</code>	<a href="#">error in:title,body</a> matches issues with "error" in their title or body.
<code>in:comments</code>	<a href="#">shipit in:comments</a> matches issues mentioning "shipit" in their comments.

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



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

If you have access to pull requests in more than 10,000 repositories, you will need to limit your search to a specific organization, personal account, or repository to see results.

Qualifier	Example
<code>user:USERNAME</code>	<a href="#">user:defunkt ubuntu</a> matches issues with the word "ubuntu" from repositories owned by @defunkt.
<code>org:ORGNAME</code>	<a href="#">org:github</a> matches issues in repositories owned by the GitHub organization.
<code>repo:USERNAME/REPOSITORY</code>	<a href="#">repo:mozilla/shumway created:&lt;2012-03-01</a> matches issues from @mozilla's shumway project that were created before March 2012.

## Search by open or closed state

You can filter issues and pull requests based on whether they're open or closed using the `state` or `is` qualifier.

Qualifier	Example
<code>state:open</code>	<a href="#">libraries state:open mentions:vmg</a> matches open issues that mention @vmg with the word "libraries."
<code>state:closed</code>	<a href="#">design state:closed in:body</a> matches closed issues with the word "design" in the body.
<code>is:open</code>	<a href="#">performance is:open is:issue</a> matches open issues with the word "performance."
<code>is:closed</code>	<a href="#">android is:closed</a> matches closed issues and pull requests with the word "android."

## Search by the reason an issue was closed

You can filter issues based on the reason given when the issue was closed, using the `reason` qualifier.

Qualifier	Example
<code>reason:completed</code>	<a href="#">libraries is:closed reason:completed</a> matches issues with the word "libraries" that were closed as "completed."
<code>reason:"not planned"</code>	<a href="#">libraries is:closed reason:"not planned"</a> matches issues with the word "libraries" that were closed as "not planned."

## Filter by repository visibility

You can filter by the visibility of the repository containing the issues and pull requests using the `is` qualifier. For more information, see "[About repositories](#)."

Qualifier	Example
<code>is:public</code>	<a href="#">is:public</a> matches issues and pull requests in public repositories.
<code>is:internal</code>	<a href="#">is:internal</a> matches issues and pull requests in internal repositories.
<code>is:private</code>	<a href="#">is:private cupcake</a> matches issues and pull requests that contain the word "cupcake" in private repositories you can access.

## Search by author

The `author` qualifier finds issues and pull requests created by a certain user or integration account.

Qualifier	Example
<code>author:USERNAME</code>	<a href="#">cool author:gjtorikian</a> matches issues and pull requests with the word "cool" that were created by @gjtorikian.
<code>in:body author:USERNAME</code>	<a href="#">bootstrap in:body author:mdo</a> matches issues written by @mdo that contain the word "bootstrap" in the body.
<code>author:app/USERNAME</code>	<a href="#">author:app/robot</a> matches issues created by the integration account named "robot."
<code>- author:app/USERNAME</code>	<a href="#">-author:app/robot</a> matches issues created by any user other than the integration account named "robot." The minus sign, or dash character ( <code>-</code> ) before the qualifier signifies a logical NOT for the qualifier in the search query.

## Search by assignee

The `assignee` qualifier finds issues and pull requests that are assigned to a certain user. You cannot search for issues and pull requests that have *any* assignee, however, you can search for [issues and pull requests that have no assignee](#).

Qualifier	Example
<code>assignee:USERNAME</code>	<a href="#">assignee:vmg repo:libgit2/libgit2</a> matches issues and pull requests in libgit2's project libgit2 that are assigned to @vmg.

## Search by mention

The `mentions` qualifier finds issues that mention a certain user. For more information, see "[Basic writing and formatting syntax](#)."

Qualifier	Example
<code>mentions:USERNAME</code>	<a href="#">resque mentions:defunkt</a> matches issues with the word "resque" that mention @defunkt.

## Search by team mention [↗](#)

For organizations and teams you belong to, you can use the `team` qualifier to find issues or pull requests that @mention a certain team within that organization. Replace these sample names with your organization and team name to perform a search.

Qualifier	Example
<code>team:ORGNAME/TEAMNAME</code>	<a href="#">team:jeekyll/owners</a> matches issues where the @jeekyll/owners team is mentioned.
<code>team:ORGNAME/TEAMNAME is:open is:pr</code>	<a href="#">team:myorg/ops is:open is:pr</a> matches open pull requests where the @myorg/ops team is mentioned.

## Search by commenter [↗](#)

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

Qualifier	Example
<code>commenter:USERNAME</code>	<a href="#">github commenter:defunkt org:github</a> matches issues in repositories owned by GitHub, that contain the word "github," and have a comment by @defunkt.

## Search by a user that's involved in an issue or pull request [↗](#)

You can use the `involves` qualifier to find issues that in some way involve a certain user. The `involves` qualifier is a logical OR between the `author`, `assignee`, `mentions`, and `commenter` qualifiers for a single user. In other words, this qualifier finds issues and pull requests that were either created by a certain user, assigned to that user, mention that user, or were commented on by that user.

Qualifier	Example
<code>involves:USERNAME</code>	<a href="#">involves:defunkt involves:jlord</a> matches issues either @defunkt or @jlord are involved in.
<code>in:body involves:USERNAME</code>	<a href="#">NOT bootstrap in:body involves:mdo</a> matches issues @mdo is involved in that do not contain the word "bootstrap" in the body.

## Search for linked issues and pull requests [↗](#)

You can narrow your results to only include issues that are linked to a pull request by a closing reference, or pull requests that are linked to an issue that the pull request may close.

Qualifier	Example
<code>linked:pr</code>	<a href="#">repo:desktop/desktop is:open linked:pr</a> matches open issues in the <code>desktop/desktop</code> repository that are linked to a pull request by a closing reference.
<code>linked:issue</code>	<a href="#">repo:desktop/desktop is:closed linked:issue</a> matches closed pull requests in the <code>desktop/desktop</code> repository that were linked to an issue that the pull request may have closed.
<code>-linked:pr</code>	<a href="#">repo:desktop/desktop is:open -linked:pr</a> matches open issues in the <code>desktop/desktop</code> repository that are not linked to a pull request by a closing reference.
<code>-linked:issue</code>	<a href="#">repo:desktop/desktop is:open -linked:issue</a> matches open pull requests in the <code>desktop/desktop</code> repository that are not linked to an issue that the pull request may close.

## Search by label

You can narrow your results by labels, using the `label` qualifier. Since issues can have multiple labels, you can list a separate qualifier for each issue.

Qualifier	Example
<code>label: LABEL</code>	<a href="#">label:"help wanted" language:ruby</a> matches issues with the label "help wanted" that are in Ruby repositories.
<code>in:body -label: LABEL label: LABEL</code>	<a href="#">broken in:body -label:bug label:priority</a> matches issues with the word "broken" in the body, that lack the label "bug", but <i>do</i> have the label "priority."
<code>label: LABEL label: LABEL</code>	<a href="#">label:bug label:resolved</a> matches issues with the labels "bug" and "resolved."
<code>label: LABEL, LABEL</code>	<a href="#">label:bug,resolved</a> matches issues with the label "bug" or the label "resolved."

## Search by milestone

The `milestone` qualifier finds issues or pull requests that are a part of a [milestone](#) within a repository.

Qualifier	Example
<code>milestone: MILESTONE</code>	<a href="#">milestone:"overhaul"</a> matches issues that are in a milestone named "overhaul."
<code>milestone: MILESTONE</code>	<a href="#">milestone:"bug fix"</a> matches issues that are in

a milestone named "bug fix."

## Search by project board

You can use the `project` qualifier to find issues that are associated with a specific [project board](#) in a repository or organization. You must search project boards by the project board number. You can find the project board number at the end of a project board's URL.

Qualifier	Example
<code>project:PROJECT_BOARD</code>	<b>project:github/57</b> matches issues owned by GitHub that are associated with the organization's project board 57.
<code>project:REPOSITORY/PROJECT_BOARD</code>	<b>project:github-linguist/linguist/1</b> matches issues that are associated with project board 1 in @github's linguist repository.

## Search by commit status

You can filter pull requests based on the status of the commits. This is especially useful if you are using [the Status API](#) or a CI service.

Qualifier	Example
<code>status:pending</code>	<b>language:go status:pending</b> matches pull requests opened into Go repositories where the status is pending.
<code>status:success</code>	<b>is:open status:success finally in:body</b> matches open pull requests with the word "finally" in the body with a successful status.
<code>status:failure</code>	<b>created:2015-05-01..2015-05-30 status:failure</b> matches pull requests opened on May 2015 with a failed status.

## Search by commit SHA

If you know the specific SHA hash of a commit, you can use it to search for pull requests that contain that SHA. The SHA syntax must be at least seven characters.

Qualifier	Example
<code>SHA</code>	<b>e1109ab</b> matches pull requests with a commit SHA that starts with <code>e1109ab</code> .
<code>SHA is:merged</code>	<b>0eff326d6213c is:merged</b> matches merged pull requests with a commit SHA that starts with <code>0eff326d6213c</code> .

## Search by branch name

You can filter pull requests based on the branch they came from (the "head" branch) or the branch they are merging into (the "base" branch).

Qualifier	Example
head: <i>HEAD_BRANCH</i>	<a href="#">head:change is:closed is:unmerged</a> matches pull requests opened from branch names beginning with the word "change" that are closed.
base: <i>BASE_BRANCH</i>	<a href="#">base:gh-pages</a> matches pull requests that are being merged into the <code>gh-pages</code> branch.

## Search by language

With the `language` qualifier you can search for issues and pull requests within repositories that are written in a certain language.

Qualifier	Example
language: <i>LANGUAGE</i>	<a href="#">language:ruby state:open</a> matches open issues that are in Ruby repositories.

## 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.

Qualifier	Example
comments: <i>n</i>	<a href="#">state:closed comments:&gt;100</a> matches closed issues with more than 100 comments.
comments: <i>n..n</i>	<a href="#">comments:500..1000</a> matches issues with comments ranging from 500 to 1,000.

## Search by number of interactions

You can filter issues and pull requests by the number of interactions with the `interactions` qualifier along with [greater than, less than, and range qualifiers](#). The interactions count is the number of reactions and comments on an issue or pull request.

Qualifier	Example
interactions: <i>n</i>	<a href="#">interactions:&gt;2000</a> matches pull requests or issues with more than 2000 interactions.
interactions: <i>n..n</i>	<a href="#">interactions:500..1000</a> matches pull requests or issues with interactions ranging from 500 to 1,000.

## Search by number of reactions

You can filter issues and pull requests by the number of reactions using the `reactions`



qualifier along with [greater than, less than, and range qualifiers](#).

Qualifier	Example
<code>reactions:n</code>	<a href="#">reactions:&gt;1000</a> matches issues with more than 1000 reactions.
<code>reactions:n..n</code>	<a href="#">reactions:500..1000</a> matches issues with reactions ranging from 500 to 1,000.

## Search for draft pull requests [↗](#)

You can filter for draft pull requests. For more information, see "[About pull requests](#)."

Qualifier	Example
<code>draft:true</code>	<a href="#">draft:true</a> matches draft pull requests.
<code>draft:false</code>	<a href="#">draft:false</a> matches pull requests that are ready for review.

## Search by pull request review status and reviewer [↗](#)

You can filter pull requests based on their [review status](#) (*none*, *required*, *approved*, or *changes requested*), by reviewer, and by requested reviewer.

Qualifier	Example
<code>review:none</code>	<a href="#">type:pr review:none</a> matches pull requests that have not been reviewed.
<code>review:required</code>	<a href="#">type:pr review:required</a> matches pull requests that require a review before they can be merged.
<code>review:approved</code>	<a href="#">type:pr review:approved</a> matches pull requests that a reviewer has approved.
<code>review:changes_requested</code>	<a href="#">type:pr review:changes_requested</a> matches pull requests in which a reviewer has asked for changes.
<code>reviewed-by:USERNAME</code>	<a href="#">type:pr reviewed-by:gjtorikian</a> matches pull requests reviewed by a particular person.
<code>review-requested:USERNAME</code>	<a href="#">type:pr review-requested:benbalter</a> matches pull requests where a specific person is requested for review. Requested reviewers are no longer listed in the search results after they review a pull request. If the requested person is on a team that is requested for review, then review requests for that team will also appear in the search results.
<code>user-review-requested:@me</code>	<a href="#">type:pr user-review-requested:@me</a> matches pull requests that you have directly been asked to review.
<code>team-review-requested:TEAMNAME</code>	<a href="#">type:pr team-review-requested:aithub/docs</a> matches pull requests

`request:github/2011-01-01` matches pull requests that have review requests from the team `github/docs`. Requested reviewers are no longer listed in the search results after they review a pull request.

## Search by when an issue or pull request was created or last updated

You can filter issues based on times of creation, or when they were last updated. For issue creation, you can use the `created` qualifier; to find out when an issue was last updated, you'll want to use the `updated` qualifier.

Both 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
<code>created:YYYY-MM-DD</code>	<a href="#">language:c# created:&lt;2011-01-01 state:open</a> matches open issues that were created before 2011 in repositories written in C#.
<code>updated:YYYY-MM-DD</code>	<a href="#">weird in:body updated:&gt;=2013-02-01</a> matches issues with the word "weird" in the body that were updated after February 2013.

## Search by when an issue or pull request was closed

You can filter issues and pull requests based on when they were closed, using the `closed` qualifier.

This qualifier takes a date as its 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
<code>closed:&gt;YYYY-MM-DD</code>	<a href="#">language:swift closed:&gt;2014-06-11</a> matches issues and pull requests in Swift that were closed after June 11, 2014.
<code>in:body closed:&lt;YYYY-MM-DD</code>	<a href="#">data in:body closed:&lt;2012-10-01</a> matches issues and pull requests with the word "data" in the body that were closed before October 2012.

## Search by when a pull request was merged

You can filter pull requests based on when they were merged, using the `merged` qualifier.

This qualifier takes a date as its 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
<code>language:LANGUAGE merged:&lt;YYYY-MM-DD</code>	<a href="#">language:javascript merged:&lt;2011-01-01</a> matches pull requests in JavaScript repositories that were merged before 2011.
<code>in:title language:LANGUAGE merged:&gt;YYYY-MM-DD</code>	<a href="#">fast in:title language:ruby merged:&gt;=2014-05-01</a> matches pull requests in Ruby with the word "fast" in the title that were merged after May 2014.

## Search based on whether a pull request is merged or unmerged [↗](#)

You can filter pull requests based on whether they're merged or unmerged using the `is` qualifier.

Qualifier	Example
<code>is:merged</code>	<a href="#">bug is:pr is:merged</a> matches merged pull requests with the word "bug."
<code>is:unmerged</code>	<a href="#">error is:unmerged</a> matches pull requests with the word "error" that are either open or were closed without being merged.

## Search based on whether a repository is archived [↗](#)

The `archived` qualifier filters your results based on whether an issue or pull request is in an archived repository.

Qualifier	Example
<code>archived:true</code>	<a href="#">archived:true GNOME</a> matches issues and pull requests that contain the word "GNOME" in archived repositories you have access to.
<code>archived:false</code>	<a href="#">archived:false GNOME</a> matches issues and pull requests that contain the word "GNOME" in unarchived repositories you have access to.

## Search based on whether a conversation is locked [↗](#)

You can search for an issue or pull request that has a locked conversation using the `is` qualifier. For more information, see "[Locking conversations](#)."

Qualifier	Example
<code>is:locked</code>	<a href="#"><b>code of conduct is:locked is:issue archived:false</b></a> matches issues or pull requests with the words "code of conduct" that have a locked conversation in a repository that is not archived.
<code>is:unlocked</code>	<a href="#"><b>code of conduct is:unlocked is:issue archived:false</b></a> matches issues or pull requests with the words "code of conduct" that have an unlocked conversation in a repository that is not archived.

## Search by missing metadata

You can narrow your search to issues and pull requests that are missing certain metadata, using the `no` qualifier. That metadata includes:

- Labels
- Milestones
- Assignees
- Projects

Qualifier	Example
<code>no:label</code>	<a href="#"><b>priority no:label</b></a> matches issues and pull requests with the word "priority" that also don't have any labels.
<code>no:milestone</code>	<a href="#"><b>sprint no:milestone type:issue</b></a> matches issues not associated with a milestone containing the word "sprint."
<code>no:assignee</code>	<a href="#"><b>important no:assignee language:java type:issue</b></a> matches issues not associated with an assignee, containing the word "important," and in Java repositories.
<code>no:project</code>	<a href="#"><b>build no:project</b></a> matches issues not associated with a project board, containing the word "build."

## Further reading

- ["Sorting search results"](#)

### Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)