Searching for repositories

You can search for repositories on GitHub and narrow the results using these repository search qualifiers in any combination.

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

To include forks in the search results, you will need to add `fork:true` or `fork:only` to your query. For more information, see "Searching in forks."

> Tips:
>
> - 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.

In this article:

Search by repository name, description, or contents of the README file

With the `in` qualifier you can restrict your search to the repository name, repository description, contents of the README file, or any combination of these. When you omit this qualifier, only the repository name and description are searched.

Qualifier	Example
`in:name`	**jquery in:name** matches repositories with "jquery" in their name.
`in:description`	**jquery in:name,description** matches repositories with "jquery" in their name or description.
`in:readme`	**jquery in:readme** matches repositories mentioning "jquery" in their README file.

Search based on the contents of a repository

You can find a repository by searching for content in its README file, using the `in:readme` qualifier.

Besides using `in:readme`, it's not possible to find repositories by searching for specific content within the repository. To search for a specific file or content within a repository, you can use the file finder or code-specific search qualifiers. For more information, see "Finding files on GitHub" and "Searching code."

Qualifier	Example
`in:readme`	**octocat in:readme** matches repositories mentioning "octocat" in their README file.

Search within a user's or organization's repositories

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

Qualifier	Example
`user: USERNAME`	**user:defunkt forks:>100** matches repositories from @defunkt that have more than 100 forks.
`org: ORGNAME`	**org:github** matches repositories from GitHub.

Search by repository size

The `size` qualifier finds repositories that match a certain size (in kilobytes), using greater than, less than, and range qualifiers.

Qualifier	Example
`size: n`	**size:1000** matches repositories that are 1 MB exactly.
	size:>=30000 matches repositories that are at least 30 MB.
	size:<50 matches repositories that are smaller than 50 KB.
	size:50..120 matches repositories that are between 50 KB and 120 KB.

Search by number of forks

The `forks` qualifier specifies the number of forks a repository should have, using greater than, less than, and range qualifiers.

Qualifier	Example
`forks: n`	**forks:5** matches repositories with only five forks.
	forks:>=205 matches repositories with at least 205 forks.
	forks:<90 matches repositories with fewer than 90 forks.
	forks:10..20 matches repositories with 10 to 20 forks.

Search by number of stars

You can search repositories based on the number of stars a repository has, using greater than, less than, and range qualifiers

Qualifier	Example
`stars: n`	**stars:500** matches repositories with exactly 500 stars.
	stars:10..20 matches repositories 10 to 20 stars, that are smaller than 1000 KB.
	stars:>=500 fork:true language:php matches repositories with the at least 500 stars, including forked ones, that are written in PHP.

Search by when a repository was created or last updated

You can filter repositories based on time of creation or time of last update. For repository creation, you can use the `created` qualifier; to find out when a repository was last updated, you'll want to use the `pushed` qualifier. The `pushed` qualifier will return a list of repositories, sorted by the most recent commit made on any branch in the repository.

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`).

Dates support greater than, less than, and range qualifiers.

Qualifier	Example
`created: YYYY-MM-DD`	**webos created:<2011-01-01** matches repositories with the word "webos" that were created before 2011.
`pushed: YYYY-MM-DD`	**css pushed:>2013-02-01** matches repositories with the word "css" that were pushed to after January 2013.
	case pushed:>=2013-03-06 fork:only matches repositories with the word "case" that were pushed to on or after March 6th, 2013, and that are forks.

Search by language

You can search repositories based on the main language they're written in.

Qualifier	Example
`language: LANGUAGE`	**rails language:javascript** matches repositories with the word "rails" that are written in JavaScript.

Search by topic

You can find all of the repositories that are classified with a particular topic.

Qualifier	Example
`topic: TOPIC`	**topic:jekyll** matches repositories that have been classified with the topic "jekyll."

Search by number of topics

You can search repositories by the number of topics that have been applied to them, using the `topics` qualifier along with greater than, less than, and range qualifiers.

Qualifier	Example
`topics: n`	**topics:5** matches repositories that have five topics.
	topics:>3 matches repositories that have more than three topics.

Search by license

You can search repositories by their license. You must use a license keyword to filter repositories by a particular license or license family.

Qualifier	Example
`license: LICENSE_KEYWORD`	**license:apache-2.0** matches repositories that are licensed under Apache License 2.0.

Search by public or private repository

You can filter your search based on whether a repository is public or private.

Qualifier	Example
`is:public`	**is:public org:github** matches repositories owned by GitHub that are public.
`is:private`	**is:private pages** matches private repositories you have access to and that contain the word "pages."

Search based on whether a repository is a mirror

You can search repositories based on whether or not they're a mirror and are hosted elsewhere.

Qualifier	Example
`mirror:true`	**mirror:true GNOME** matches repositories that are mirrors and contain the word "GNOME."
`mirror:false`	**mirror:false GNOME** matches repositories that are not mirrors and contain the word "GNOME."

Search based on whether a repository is archived

You can search repositories based on whether or not they're archived.

Qualifier	Example

Qualifier	Example
`archived:true`	archived:true GNOME matches repositories that are archived and contain the word "GNOME."
`archived:false`	archived:false GNOME matches repositories that are not archived and contain the word "GNOME."

Search based on number of issues with `good first issue` or `help wanted` labels

You can search for repositories that have a minimum number of issues labeled `help-wanted` or `good-first-issue` with the qualifiers `help-wanted-issues:>n` and `good-first-issues:>n`. For more information, see "Helping new contributors find your project with labels."

Qualifier	Example
`good-first-issues:>n`	good-first-issues:>2 javascript matches repositories with more than two issues labeled `good-first-issue` and that contain the word "javascript."
`help-wanted-issues:>n`	help-wanted-issues:>4 react matches repositories with more than four issues labeled `help-wanted` and that contain the word "React."

Further reading

- "Sorting search results"
- "Searching in forks"