Searching code (legacy)

In this article

You only need to use the legacy code search syntax if you are using the code search API.

> **Note:** This article covers the syntax for legacy code search, which you should only need to use for the [REST API endpoint for searching code](#).
>
> For information on the code search syntax that you can use on GitHub.com, see "[Understanding GitHub Code Search syntax](#)."

You can search globally across all of GitHub Enterprise Cloud, or scope your search to a particular repository or organization. For more information, see "[About searching on GitHub](#)."

You can only search code using these code search qualifiers. Search qualifiers specifically for repositories, users, or commits, will not work when searching for code.

> **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.

Considerations for code search 🔗

Due to the complexity of searching code, there are some restrictions on how searches are performed:

- You must be signed into a personal account on GitHub Enterprise Cloud to search for code across all public repositories.
- Code in [forks](#) is only searchable if the fork has more stars than the parent repository. Forks with fewer stars than the parent repository are **not** indexed for code search. To include forks with more stars than their parent in the search results, you will need to add `fork:true` or `fork:only` to your query. For more information, see "[Searching in forks](#)."

- Only the *default branch* is indexed for code search.
- Only files smaller than 384 KB are searchable.
- Up to 4,000 private and internal repositories are searchable. These 4,000 repositories will be the most recently updated of the first 10,000 private and internal repositories that you have access to.
- Only repositories with fewer than 500,000 files are searchable.
- Only repositories that have had activity or have been returned in search results in the last year are searchable.
- Except with `filename` searches, you must always include at least one search term when searching source code. For example, searching for `language:javascript` is not valid, while `amazing language:javascript` is.
- At most, search results can show two fragments from the same file, but there may be more results within the file.
- You can't use the following wildcard characters as part of your search query: `. , : ; / \ ` ' " = * ! ? # $ & + ^ | ~ < > ( ) { } [ ] @`. The search will simply ignore these symbols.

Search by the file contents or file path 🔗

With the `in` qualifier you can restrict your search to the contents of the source code file, the file path, or both. When you omit this qualifier, only the file contents are searched.

Qualifier	Example
`in:file`	**octocat in:file** matches code where "octocat" appears in the file contents.
`in:path`	**octocat in:path** matches code where "octocat" appears in the file path.
`in:file,path`	**octocat in:file,path** matches code where "octocat" appears in the file contents or the file path.

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

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

Qualifier	Example
`user:USERNAME`	**user:defunkt extension:rb** matches code from @defunkt that ends in *.rb*.
`org:ORGNAME`	**org:github extension:js** matches code from GitHub that ends in *.js*.
`repo:USERNAME/REPOSITORY`	**repo:mozilla/shumway extension:as** matches code from @mozilla's shumway project that ends in *.as*.

Search by file location 🔗

You can use the `path` qualifier to search for source code that appears at a specific location in a repository. Use `path:/` to search for files that are located at the root level of a repository. Or specify a directory name or the path to a directory to search for files that are located within that directory or any of its subdirectories.

Qualifier	Example
`path:/`	**octocat filename:readme path:/** matches *readme* files with the word "octocat" that are located at the root level of a repository.
`path:DIRECTORY`	**form path:cgi-bin language:perl** matches Perl files with the word "form" in the *cgi-bin* directory, or in any of its subdirectories.
`path:PATH/TO/DIRECTORY`	`console path:app/public language:javascript` matches JavaScript files with the word "console" in the *app/public* directory, or in any of its subdirectories (even if they reside in *app/public/js/form-validators*).

Search by language 🔗

You can search for code based on what language it's written in. The `language` qualifier can be the language name or alias. For a full list of supported languages with their names and aliases, see the [github-linguist/linguist repository](#).

Qualifier	Example
`language:LANGUAGE`	**element language:xml size:100** matches code with the word "element" that's marked as being XML and has exactly 100 bytes.
`language:LANGUAGE`	**display language:scss** matches code with the word "display," that's marked as being SCSS.
`language:LANGUAGE`	**org:mozilla language:markdown** matches code from all @mozilla's repositories that's marked as Markdown.

Search by file size 🔗

You can use the `size` qualifier to search for source code based on the size of the file where the code exists. The `size` qualifier uses [greater than, less than, and range qualifiers](#) to filter results based on the byte size of the file in which the code is found.

Qualifier	Example
`size:n`	**function size:>10000 language:python** matches code with the word "function," written in Python, in files that are larger than 10 KB.

Search by filename 🔗

The `filename` qualifier matches code files with a certain filename. You can also find a file in a repository using the file finder. For more information, see "[Finding files on GitHub](#)."

Qualifier	Example
`filename:`*FILENAME*	**filename:linguist** matches files named "linguist."
`filename:`*FILENAME*	**filename:.vimrc commands** matches *.vimrc* files with the word "commands."
`filename:`*FILENAME*	**filename:test_helper path:test language:ruby** matches Ruby files named *test_helper* within the *test* directory.

Search by file extension 🔗

The `extension` qualifier matches code files with a certain file extension.

Qualifier	Example
`extension:`*EXTENSION*	**form path:cgi-bin extension:pm** matches code with the word "form," under *cgi-bin*, with the *.pm* file extension.
`extension:`*EXTENSION*	**icon size:>200000 extension:css** matches files larger than 200 KB that end in .css and have the word "icon."

Further reading 🔗

- "Sorting search results"
- "Searching in forks"
- "Navigating code on GitHub"