
beagle Documentation

Doug Hellmann

Jul 07, 2020

Contents

1	beagle installation guide	3
2	Command line interface reference	5
2.1	search	5
3	Users guide	7
3.1	Examples	7
3.2	OpenStack Client Integration	9
4	CHANGES	11
4.1	0.3.0	11
4.2	0.2.2	11
4.3	0.2.1	11
4.4	0.2.0	12
4.5	0.1.0	12
5	Indices and tables	13
	Index	15

Beagle is a command line client for [Hound](#), the code search tool.

Source <https://github.com/beaglecli/beagle>

Bugs <https://github.com/beaglecli/beagle/issues>

CHAPTER 1

beagle installation guide

The beagle package should be installed via pip:

```
$ pip install beagle
```

Command line interface reference

2.1 search

Search for a pattern and show the results.

```
beagle search
  [-f {csv,grep,json,link,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--ignore-comments]
  [--comment-marker COMMENT_MARKER]
  [--file FILE_PATTERN]
  [--ignore-case]
  [--repo REPOS]
  [--repo-pattern REPO_PATTERN]
  [--context-lines CONTEXT_LINES]
  query
```

-f <FORMATTER>, **--format** <FORMATTER>
the output format, defaults to table

-c COLUMN, **--column** COLUMN
specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>
when to include quotes, defaults to nonnumeric

--noindent
whether to disable indenting the JSON

--max-width <integer>
Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width
Fit the table to the display width. Implied if **--max-width** greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty
Print empty table if there is no data to show.

--sort-column SORT_COLUMN
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--ignore-comments
ignore comment lines

--comment-marker <COMMENT_MARKER>
start of a comment line

--file <FILE_PATTERN>, **--file-pattern** <FILE_PATTERN>
file name pattern

--ignore-case
ignore case in search string

--repo <REPOS>
limit search to named repositories (option can be repeated)

--repo-pattern <REPO_PATTERN>
glob pattern to match repository names (filtered on client side)

--context-lines <CONTEXT_LINES>
number of context lines

query
the text pattern

This command is provided by the beagle plugin.

3.1 Examples

To find requirements or constraints specifications for `cliff`:

```
$ beagle search --file '(*.requirement.*|*.constraint.*|setup.cfg|tox.ini)' 'cliff[>
↳<=]'
```

Repository	Filename
group-based-policy	test-requirements.txt
20 cliff>=2.3.0 # Apache-2.0	
kolla-kubernetes	requirements.txt
6 cliff>=2.8.0 # Apache-2.0	
networking-bigswitch	test-requirements.txt
8 cliff>=1.7.0 # Apache-2.0	
networking-brocade	test-requirements.txt
6 cliff>=1.14.0 # Apache-2.0	
networking-mlnx	test-requirements.txt
6 cliff>=1.15.0 # Apache-2.0	
networking-plumgrid	test-requirements.txt
3 cliff>=2.2.0 # Apache-2.0	
osops-tools-contrib	ansible_requirements.txt
5 cliff==2.2.0	
paunch	requirements.txt
7 cliff>=2.6.0 # Apache-2.0	
rally	upper-constraints.txt
68 cliff===2.11.0	
requirements	global-requirements.txt
21 cliff>=2.8.0,!>=2.9.0 # Apache-2.0	

(continues on next page)

(continued from previous page)

```
| requirements | openstack_requirements/tests/files/gr-base.txt |
↪| 9 | cliff>=1.4 |
| requirements | openstack_requirements/tests/files/upper-constraints.txt |
↪| 192 | cliff===2.4.0 |
| requirements | upper-constraints.txt |
↪| 216 | cliff===2.11.0 |
| rpm-packaging | requirements.txt |
↪| 21 | cliff>=2.8.0,!>=2.9.0 # Apache-2.0 |
+-----+-----+
↪+-----+-----+
```

To show the 5 lines before and after the location of the class definition for `ConfigOpts` in `oslo.config` using the `grep` output formatter.

```
$ beagle search -f grep --context-lines 5 --repo openstack/oslo.config 'class_
↪ConfigOpts'
openstack/oslo.config:doc/source/reference/configuration-files.rst:5:The config_
↪manager has two CLI options defined by default, ``--config-file``
openstack/oslo.config:doc/source/reference/configuration-files.rst:6:and ``--config-
↪dir``:
openstack/oslo.config:doc/source/reference/configuration-files.rst:7:
openstack/oslo.config:doc/source/reference/configuration-files.rst:8:... code-block::_
↪python
openstack/oslo.config:doc/source/reference/configuration-files.rst:9:
openstack/oslo.config:doc/source/reference/configuration-files.rst:10:class_
↪ConfigOpts(object):
openstack/oslo.config:doc/source/reference/configuration-files.rst:11:
openstack/oslo.config:doc/source/reference/configuration-files.rst:12:         def __
↪call__(self, ...):
openstack/oslo.config:doc/source/reference/configuration-files.rst:13:
openstack/oslo.config:doc/source/reference/configuration-files.rst:14:         _
↪opts = [
openstack/oslo.config:doc/source/reference/configuration-files.rst:15:         _
↪MultiStrOpt('config-file',
openstack/oslo.config:oslo_config/cfg.py:1920:         def print_usage(self, file=None):
openstack/oslo.config:oslo_config/cfg.py:1921:             self.initialize_parser_
↪arguments()
openstack/oslo.config:oslo_config/cfg.py:1922:             super(_CachedArgumentParser,_
↪self).print_usage(file)
openstack/oslo.config:oslo_config/cfg.py:1923:
openstack/oslo.config:oslo_config/cfg.py:1924:
openstack/oslo.config:oslo_config/cfg.py:1925:class ConfigOpts(abc.Mapping):
openstack/oslo.config:oslo_config/cfg.py:1926:
openstack/oslo.config:oslo_config/cfg.py:1927:     """Config options which may be set_
↪on the command line or in config files.
openstack/oslo.config:oslo_config/cfg.py:1928:
openstack/oslo.config:oslo_config/cfg.py:1929:     ConfigOpts is a configuration_
↪option manager with APIs for registering
openstack/oslo.config:oslo_config/cfg.py:1930:     option schemas, grouping options,_
↪parsing option values and retrieving
```

To produce links to the source on the OpenStack git server, use the link formatter:

```
$ beagle --debug search -f link --repo openstack/oslo.config 'class ConfigOpts'
https://opendev.org/openstack/oslo.config/src/branch/master/doc/source/reference/
↪configuration-files.rst#n10 : class ConfigOpts(object):
```

(continues on next page)

(continued from previous page)

```
https://opendev.org/openstack/oslo.config/src/branch/master/oslo_config/cfg.py#n1925_
↪: class ConfigOpts(abc.Mapping):
```

To filter repositories in search results, use the `--repo-pattern` option.

Example to show which openstack oslo project call the `ssl.wrap_socket` function:

```
$ beagle search --ignore-comments -f link --repo-pattern "openstack/oslo.*" 'ssl.wrap_
↪socket'
https://opendev.org/openstack/oslo.service/src/branch/master/oslo_service/sslutils.py
↪#n104 : return ssl.wrap_socket(sock, **ssl_kwargs) # nosec
https://opendev.org/openstack/oslo.service/src/branch/master/oslo_service/tests/test_
↪sslutils.py#n81 : @mock.patch("ssl.wrap_socket")
```

Same research only in the whole openstack projects (will ignore starlingx, etc):

```
$ beagle search --ignore-comments -f link --repo-pattern "openstack/*" 'ssl.wrap_
↪socket'
https://opendev.org/openstack/glance/src/branch/master/glance/common/client.py#n124 : ↪
↪ssl.wrap_socket(), which forces SSL to check server certificate against
https://opendev.org/openstack/glance/src/branch/master/glance/common/client.py#n133 : ↪
↪self.sock = ssl.wrap_socket(sock, self.key_file, self.cert_file,
https://opendev.org/openstack/glance/src/branch/master/glance/common/client.py#n136 : ↪
↪self.sock = ssl.wrap_socket(sock, self.key_file, self.cert_file,
https://opendev.org/openstack/heat/src/branch/master/heat/common/wsgi.py#n239 : ssl.
↪wrap_socket if conf specifies cert_file
https://opendev.org/openstack/heat/src/branch/master/heat/common/wsgi.py#n414 : self.
↪sock = ssl.wrap_socket(self._sock,
```

3.2 OpenStack Client Integration

When the `python-openstackclient` package and `beagle` are both installed, it is possible to search the OpenStack source code directly from the `openstack` command line tool.

```
$ openstack code search --file '(.*requirement.*|.*constraint.*|setup.cfg|tox.ini)'
↪'cliff[>=]'
```

Repository	Filename
group-based-policy	test-requirements.txt
20	cliff>=2.3.0 # Apache-2.0
kolla-kubernetes	requirements.txt
6	cliff>=2.8.0 # Apache-2.0
networking-bigswitch	test-requirements.txt
8	cliff>=1.7.0 # Apache-2.0
networking-brocade	test-requirements.txt
6	cliff>=1.14.0 # Apache-2.0
networking-mlnx	test-requirements.txt
6	cliff>=1.15.0 # Apache-2.0
networking-plumgrid	test-requirements.txt
3	cliff>=2.2.0 # Apache-2.0

(continues on next page)

(continued from previous page)

```
| osops-tools-contrib | ansible_requirements.txt |
↪| 5 | cliff==2.2.0 |
| paunch | requirements.txt |
↪| 7 | cliff>=2.6.0 # Apache-2.0 |
| rally | upper-constraints.txt |
↪| 68 | cliff===2.11.0 |
| requirements | global-requirements.txt |
↪| 21 | cliff>=2.8.0,! =2.9.0 # Apache-2.0 |
| requirements | openstack_requirements/tests/files/gr-base.txt |
↪| 9 | cliff>=1.4 |
| requirements | openstack_requirements/tests/files/upper-constraints.txt |
↪| 192 | cliff===2.4.0 |
| requirements | upper-constraints.txt |
↪| 216 | cliff===2.11.0 |
| rpm-packaging | requirements.txt |
↪| 21 | cliff>=2.8.0,! =2.9.0 # Apache-2.0 |
+-----+-----+
↪+-----+-----+
```

4.1 0.3.0

- add new rules to gitignore
- Add repo-pattern usages examples in the doc
- filter repositories in search results
- set up tests for newer versions of python
- Refresh python's versions and their usages
- Redirect project's home page

4.2 0.2.2

- add settings so travis can publish releases
- update urls in docs after github org move
- fix the reference to undefined function in link formatter
- Fix issues
- Refactor pipelines
- [doc] refresh oslo examples

4.3 0.2.1

- use correct form of affirmative boolean value for ignore case

4.4 0.2.0

- add link formatter for producing links to online copies of source
- add a few examples to the user docs
- add python-openstackclient integration
- ignore build directory

4.5 0.1.0

- add source and doc links to the readme
- fix the documentation build
- pep8 fixes
- use formattable column for before and after
- refactor before/after handling in grep output
- write grep formatter output to stdout handle given
- add a pseudo-grep output formatter
- ignore coverage output directory
- add `-context-lines` option
- add options for controlling query parameters
- remove superfluous functions from import of hound query func
- remove extra whitespace from lines in the output
- implement basic search command
- make the doc build work
- stub in basic cliff application
- add basic README to get started

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`

Symbols

-comment-marker <COMMENT_MARKER>
 beagle-search command line option, 6

-context-lines <CONTEXT_LINES>
 beagle-search command line option, 6

-file <FILE_PATTERN>, -file-pattern
 <FILE_PATTERN>
 beagle-search command line option, 6

-fit-width
 beagle-search command line option, 6

-ignore-case
 beagle-search command line option, 6

-ignore-comments
 beagle-search command line option, 6

-max-width <integer>
 beagle-search command line option, 5

-noindent
 beagle-search command line option, 5

-print-empty
 beagle-search command line option, 6

-quote <QUOTE_MODE>
 beagle-search command line option, 5

-repo <REPOS>
 beagle-search command line option, 6

-repo-pattern <REPO_PATTERN>
 beagle-search command line option, 6

-sort-column SORT_COLUMN
 beagle-search command line option, 6

-c COLUMN, -column COLUMN
 beagle-search command line option, 5

-f <FORMATTER>, -format <FORMATTER>
 beagle-search command line option, 5

B

beagle-search command line option
 -comment-marker <COMMENT_MARKER>, 6
 -context-lines <CONTEXT_LINES>, 6
 -file <FILE_PATTERN>,
 -file-pattern <FILE_PATTERN>,

6

-fit-width, 6
 -ignore-case, 6
 -ignore-comments, 6
 -max-width <integer>, 5
 -noindent, 5
 -print-empty, 6
 -quote <QUOTE_MODE>, 5
 -repo <REPOS>, 6
 -repo-pattern <REPO_PATTERN>, 6
 -sort-column SORT_COLUMN, 6
 -c COLUMN, -column COLUMN, 5
 -f <FORMATTER>, -format
 <FORMATTER>, 5
 query, 6

Q

query
 beagle-search command line option, 6