
Elasticsearch Documentation

Release 2.0.0

Honza Král

October 13, 2015

1	Compatibility	3
2	Example Usage	5
3	Features	7
3.1	Persistent Connections	7
3.2	Automatic Retries	7
3.3	Sniffing	8
3.4	SSL and Authentication	8
3.5	Logging	8
4	Environment considerations	11
5	Contents	13
5.1	API Documentation	13
5.2	Exceptions	53
5.3	Connection Layer API	53
5.4	Transport classes	57
5.5	Helpers	58
5.6	Changelog	62
6	License	65
7	Indices and tables	67
	Python Module Index	69

Official low-level client for Elasticsearch. Its goal is to provide common ground for all Elasticsearch-related code in Python; because of this it tries to be opinion-free and very extendable.

For a more high level client library with more limited scope, have a look at [elasticsearch-dsl](#) - it is a more pythonic library sitting on top of `elasticsearch-py`.

Compatibility

The library is compatible with all Elasticsearch versions since 0.90.x but you **have to use a matching major version**:

For **Elasticsearch 2.0** and later, use the major version 1 (1.x.y) of the library.

For **Elasticsearch 1.0** and later, use the major version 1 (1.x.y) of the library.

For **Elasticsearch 0.90.x**, use a version from 0.4.x releases of the library.

The recommended way to set your requirements in your *setup.py* or *requirements.txt* is:

```
# Elasticsearch 2.x
elasticsearch>=2.0.0,<3.0.0

# Elasticsearch 1.x
elasticsearch>=1.0.0,<2.0.0

# Elasticsearch 0.90.x
elasticsearch<1.0.0
```

The development is happening on master and 1.x branches, respectively.

Example Usage

```
from datetime import datetime
from elasticsearch import Elasticsearch
es = Elasticsearch()

doc = {
    'author': 'kimchy',
    'text': 'Elasticsearch: cool. bonsai cool.',
    'timestamp': datetime.now(),
}
res = es.index(index="test-index", doc_type='tweet', id=1, body=doc)
print(res['_created'])

res = es.get(index="test-index", doc_type='tweet', id=1)
print(res['_source'])

es.indices.refresh(index="test-index")

res = es.search(index="test-index", body={"query": {"match_all": {}}})
print("Got %d Hits:" % res['hits']['total'])
for hit in res['hits']['hits']:
    print("(%(timestamp)s %(author)s: %(text)s" % hit["_source"])
```

Features

This client was designed as very thin wrapper around Elasticsearch's REST API to allow for maximum flexibility. This means that there are no opinions in this client; it also means that some of the APIs are a little cumbersome to use from Python. We have created some *Helpers* to help with this issue.

3.1 Persistent Connections

`elasticsearch-py` uses persistent connections inside of individual connection pools (one per each configured or sniffed node). Out of the box you can choose to use `http`, `thrift` or an experimental `memcached` protocol to communicate with the elasticsearch nodes. See *Transport classes* for more information.

The transport layer will create an instance of the selected connection class per node and keep track of the health of individual nodes - if a node becomes unresponsive (throwing exceptions while connecting to it) it's put on a timeout by the *ConnectionPool* class and only returned to the circulation after the timeout is over (or when no live nodes are left). By default nodes are randomized before being passed into the pool and round-robin strategy is used for load balancing.

You can customize this behavior by passing parameters to the *Connection Layer API* (all keyword arguments to the *Elasticsearch* class will be passed through). If what you want to accomplish is not supported you should be able to create a subclass of the relevant component and pass it in as a parameter to be used instead of the default implementation.

Note: Since we use persistent connections throughout the client it means that the client doesn't tolerate `fork` very well. If your application calls for multiple processes make sure you create a fresh client after call to `fork`.

3.2 Automatic Retries

If a connection to a node fails due to connection issues (raises *ConnectionError*) it is considered in faulty state. It will be placed on hold for `dead_timeout` seconds and the request will be retried on another node. If a connection fails multiple times in a row the timeout will get progressively larger to avoid hitting a node that's, by all indication, down. If no live connection is available, the connection that has the smallest timeout will be used.

By default retries are not triggered by a timeout (*ConnectionTimeout*), set `retry_on_timeout` to `True` to also retry on timeouts.

3.3 Sniffing

The client can be configured to inspect the cluster state to get a list of nodes upon startup, periodically and/or on failure. See *Transport* parameters for details.

Some example configurations:

```
from elasticsearch import Elasticsearch

# by default we don't sniff, ever
es = Elasticsearch()

# you can specify to sniff on startup to inspect the cluster and load
# balance across all nodes
es = Elasticsearch(["seed1", "seed2"], sniff_on_start=True)

# you can also sniff periodically and/or after failure:
es = Elasticsearch(["seed1", "seed2"], sniff_on_start=True, sniff_on_connection_fail=True, sniffer_t
```

3.4 SSL and Authentication

You can configure the client to use SSL for connecting to your elasticsearch cluster, including certificate verification and http auth:

```
from elasticsearch import Elasticsearch

# you can use RFC-1738 to specify the url
es = Elasticsearch(['https://user:secret@localhost:443'])

# ... or specify common parameters as kwargs

# use certifi for CA certificates
import certifi

es = Elasticsearch(
    ['localhost', 'otherhost'],
    http_auth=('user', 'secret'),
    port=443,
    use_ssl=True,
    verify_certs=True,
    ca_certs=certifi.where(),
)
```

Warning: By default SSL certificates won't be verified, pass in `verify_certs=True` to make sure your certificates will get verified. The client doesn't ship with any CA certificates; easiest way to obtain the common set is by using the `certifi` package (as shown above).

See class *Urllib3HttpConnection* for detailed description of the options.

3.5 Logging

elasticsearch-py uses the standard *logging library* from python to define two loggers: `elasticsearch` and `elasticsearch.trace`. `elasticsearch` is used by the client to log standard activity, depending on the log

level. `elasticsearch.trace` can be used to log requests to the server in the form of `curl` commands using pretty-printed json that can then be executed from command line. If the trace logger has not been configured already it is set to *propagate=False* so it needs to be activated separately.

Environment considerations

When using the client there are several limitations of your environment that could come into play.

When using an http load balancer you cannot use the *Sniffing* functionality - the cluster would supply the client with IP addresses to directly connect to the cluster, circumventing the load balancer. Depending on your configuration this might be something you don't want or break completely.

In some environments (notably on Google App Engine) your http requests might be restricted so that GET requests won't accept body. In that case use the `send_get_body_as` parameter of *Transport* to send all bodies via post:

```
from elasticsearch import Elasticsearch
es = Elasticsearch(send_get_body_as='POST')
```

Contents

5.1 API Documentation

All the API calls map the raw REST api as closely as possible, including the distinction between required and optional arguments to the calls. This means that the code makes distinction between positional and keyword arguments; we, however, recommend that people **use keyword arguments for all calls for consistency and safety**.

Note: for compatibility with the Python ecosystem we use `from_` instead of `from` and `doc_type` instead of `type` as parameter names.

5.1.1 Global options

Some parameters are added by the client itself and can be used in all API calls.

Ignore

An API call is considered successful (and will return a response) if elasticsearch returns a 2XX response. Otherwise an instance of *TransportError* (or a more specific subclass) will be raised. You can see other exception and error states in *Exceptions*. If you do not wish an exception to be raised you can always pass in an `ignore` parameter with either a single status code that should be ignored or a list of them:

```
from elasticsearch import Elasticsearch
es = Elasticsearch()

# ignore 400 cause by IndexAlreadyExistsException when creating an index
es.indices.create(index='test-index', ignore=400)

# ignore 404 and 400
es.indices.delete(index='test-index', ignore=[400, 404])
```

Timeout

Global timeout can be set when constructing the client (see `Connection`'s `timeout` parameter) or on a per-request basis using `request_timeout` (float value in seconds) as part of any API call, this value will get passed to the `perform_request` method of the connection class:

```
# only wait for 1 second, regardless of the client's default
es.cluster.health(wait_for_status='yellow', request_timeout=1)
```

Note: Some API calls also accept a `timeout` parameter that is passed to Elasticsearch server. This timeout is internal and doesn't guarantee that the request will end in the specified time.

5.1.2 Elasticsearch

class `elasticsearch.Elasticsearch` (*hosts=None*, *transport_class=<class 'elasticsearch.transport.Transport'>*, ***kwargs*)

Elasticsearch low-level client. Provides a straightforward mapping from Python to ES REST endpoints.

The instance has attributes `cat`, `cluster`, `indices`, `nodes` and `snapshot` that provide access to instances of *CatClient*, *ClusterClient*, *IndicesClient*, *NodesClient* and *SnapshotClient* respectively. This is the preferred (and only supported) way to get access to those classes and their methods.

You can specify your own connection class which should be used by providing the `connection_class` parameter:

```
# create connection to localhost using the ThriftConnection
es = Elasticsearch(connection_class=ThriftConnection)
```

If you want to turn on *Sniffing* you have several options (described in *Transport*):

```
# create connection that will automatically inspect the cluster to get
# the list of active nodes. Start with nodes running on 'esnode1' and
# 'esnode2'
es = Elasticsearch(
    ['esnode1', 'esnode2'],
    # sniff before doing anything
    sniff_on_start=True,
    # refresh nodes after a node fails to respond
    sniff_on_connection_fail=True,
    # and also every 60 seconds
    sniffer_timeout=60
)
```

Different hosts can have different parameters, use a dictionary per node to specify those:

```
# connect to localhost directly and another node using SSL on port 443
# and an url_prefix. Note that ``port`` needs to be an int.
es = Elasticsearch([
    {'host': 'localhost'},
    {'host': 'othernode', 'port': 443, 'url_prefix': 'es', 'use_ssl': True},
])
```

If using SSL, there are several parameters that control how we deal with certificates (see *Urllib3HttpConnection* for detailed description of the options):

```
es = Elasticsearch(
    ['localhost:443', 'other_host:443'],
    # turn on SSL
    use_ssl=True,
    # make sure we verify SSL certificates (off by default)
    verify_certs=True,
    # provide a path to CA certs on disk
```

```

    ca_certs='/path/to/CA_certs'
)

```

Alternatively you can use RFC-1738 formatted URLs, as long as they are not in conflict with other options:

```

es = Elasticsearch(
    [
        'http://user:secret@localhost:9200/',
        'https://user:secret@other_host:443/production'
    ],
    verify_certs=True
)

```

Parameters

- **hosts** – list of nodes we should connect to. Node should be a dictionary ({“host”: “localhost”, “port”: 9200}), the entire dictionary will be passed to the `Connection` class as kwargs, or a string in the format of `host[:port]` which will be translated to a dictionary automatically. If no value is given the `Urllib3HttpConnection` class defaults will be used.
- **transport_class** – `Transport` subclass to use.
- **kwargs** – any additional arguments will be passed on to the `Transport` class and, subsequently, to the `Connection` instances.

bulk (*args, **kwargs)

Perform many index/delete operations in a single API call.

See the `bulk()` helper function for a more friendly API. http://www.elastic.co/guide/en/elasticsearch/reference/current/docs_bulk.html

Parameters

- **body** – The operation definition and data (action-data pairs), separated by newlines
- **index** – Default index for items which don’t provide one
- **doc_type** – Default document type for items which don’t provide one
- **consistency** – Explicit write consistency setting for the operation, valid choices are: ‘one’, ‘quorum’, ‘all’
- **fields** – Default comma-separated list of fields to return in the response for updates
- **refresh** – Refresh the index after performing the operation
- **routing** – Specific routing value
- **timeout** – Explicit operation timeout

clear_scroll (*args, **kwargs)

Clear the scroll request created by specifying the scroll parameter to search. <http://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-scroll.html>

Parameters

- **scroll_id** – A comma-separated list of scroll IDs to clear
- **body** – A comma-separated list of scroll IDs to clear if none was specified via the `scroll_id` parameter

count (*args, **kwargs)

Execute a query and get the number of matches for that query.
<http://www.elastic.co/guide/en/elasticsearch/reference/current/search-count.html>

Parameters

- **index** – A comma-separated list of indices to restrict the results
- **doc_type** – A comma-separated list of types to restrict the results
- **body** – A query to restrict the results specified with the Query DSL (optional)
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **analyze_wildcard** – Specify whether wildcard and prefix queries should be analyzed (default: false)
- **analyzer** – The analyzer to use for the query string
- **default_operator** – The default operator for query string query (AND or OR), default 'OR', valid choices are: 'AND', 'OR'
- **df** – The field to use as default where no field prefix is given in the query string
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open', valid choices are: 'open', 'closed', 'none', 'all'
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **lenient** – Specify whether format-based query failures (such as providing text to a numeric field) should be ignored
- **lowercase_expanded_terms** – Specify whether query terms should be lowercased
- **min_score** – Include only documents with a specific *_score* value in the result
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **q** – Query in the Lucene query string syntax
- **routing** – Specific routing value

count_percolate (*args, **kwargs)

The percolator allows to register queries against an index, and then send percolate requests which include a doc, and getting back the queries that match on that doc out of the set of registered queries.
<http://www.elastic.co/guide/en/elasticsearch/reference/current/search-percolate.html>

Parameters

- **index** – The index of the document being count percolated.
- **doc_type** – The type of the document being count percolated.
- **id** – Substitute the document in the request body with a document that is known by the specified id. On top of the id, the index and type parameter will be used to retrieve the document from within the cluster.
- **body** – The count percolator request definition using the percolate DSL
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)

- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **percolate_index** – The index to count percolate the document into. Defaults to index.
- **percolate_type** – The type to count percolate document into. Defaults to type.
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **routing** – A comma-separated list of specific routing values
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: ‘internal’, ‘external’, ‘external_gte’, ‘force’

create (*args, **kwargs)

Adds a typed JSON document in a specific index, making it searchable. Behind the scenes this method calls `index(..., op_type='create')` http://www.elastic.co/guide/en/elasticsearch/reference/current/docs-index_.html

Parameters

- **index** – The name of the index
- **doc_type** – The type of the document
- **body** – The document
- **id** – Document ID
- **consistency** – Explicit write consistency setting for the operation, valid choices are: ‘one’, ‘quorum’, ‘all’
- **op_type** – Explicit operation type, default ‘index’, valid choices are: ‘index’, ‘create’
- **parent** – ID of the parent document
- **refresh** – Refresh the index after performing the operation
- **routing** – Specific routing value
- **timeout** – Explicit operation timeout
- **timestamp** – Explicit timestamp for the document
- **t1** – Expiration time for the document
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: ‘internal’, ‘external’, ‘external_gte’, ‘force’

delete (*args, **kwargs)

Delete a typed JSON document from a specific index based on its id. <http://www.elastic.co/guide/en/elasticsearch/reference/current/docs-delete.html>

Parameters

- **index** – The name of the index
- **doc_type** – The type of the document

- **id** – The document ID
- **consistency** – Specific write consistency setting for the operation, valid choices are: ‘one’, ‘quorum’, ‘all’
- **parent** – ID of parent document
- **refresh** – Refresh the index after performing the operation
- **routing** – Specific routing value
- **timeout** – Explicit operation timeout
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: ‘internal’, ‘external’, ‘external_gte’, ‘force’

delete_script (*args, **kwargs)

Remove a stored script from elasticsearch. <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-scripting.html>

Parameters

- **lang** – Script language
- **id** – Script ID
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: ‘internal’, ‘external’, ‘external_gte’, ‘force’

delete_template (*args, **kwargs)

Delete a search template. <http://www.elastic.co/guide/en/elasticsearch/reference/current/search-template.html>

Parameters

- **id** – Template ID
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: ‘internal’, ‘external’, ‘external_gte’, ‘force’

exists (*args, **kwargs)

Returns a boolean indicating whether or not given document exists in Elasticsearch. <http://www.elastic.co/guide/en/elasticsearch/reference/current/docs-get.html>

Parameters

- **index** – The name of the index
- **doc_type** – The type of the document (use *_all* to fetch the first document matching the ID across all types)
- **id** – The document ID
- **parent** – The ID of the parent document
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **realtime** – Specify whether to perform the operation in realtime or search mode
- **refresh** – Refresh the shard containing the document before performing the operation

- **routing** – Specific routing value

explain (*args, **kwargs)

The explain api computes a score explanation for a query and a specific document. This can give useful feedback whether a document matches or didn't match a specific query. <http://www.elastic.co/guide/en/elasticsearch/reference/current/search-explain.html>

Parameters

- **index** – The name of the index
- **doc_type** – The type of the document
- **id** – The document ID
- **body** – The query definition using the Query DSL
- **_source** – True or false to return the _source field or not, or a list of fields to return
- **_source_exclude** – A list of fields to exclude from the returned _source field
- **_source_include** – A list of fields to extract and return from the _source field
- **analyze_wildcard** – Specify whether wildcards and prefix queries in the query string query should be analyzed (default: false)
- **analyzer** – The analyzer for the query string query
- **default_operator** – The default operator for query string query (AND or OR), default 'OR', valid choices are: 'AND', 'OR'
- **df** – The default field for query string query (default: _all)
- **fields** – A comma-separated list of fields to return in the response
- **lenient** – Specify whether format-based query failures (such as providing text to a numeric field) should be ignored
- **lowercase_expanded_terms** – Specify whether query terms should be lowercased
- **parent** – The ID of the parent document
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **q** – Query in the Lucene query string syntax
- **routing** – Specific routing value

field_stats (*args, **kwargs)

The field stats api allows one to find statistical properties of a field without executing a search, but looking up measurements that are natively available in the Lucene index. <http://www.elastic.co/guide/en/elasticsearch/reference/current/search-field-stats.html>

Parameters

- **index** – A comma-separated list of index names; use *_all* or empty string to perform the operation on all indices
- **body** – Field json objects containing the name and optionally a range to filter out indices result, that have results outside the defined bounds
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open', valid choices are: 'open', 'closed', 'none', 'all'

- **fields** – A comma-separated list of fields for to get field statistics for (min value, max value, and more)
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **level** – Defines if field stats should be returned on a per index level or on a cluster wide level, default ‘cluster’, valid choices are: ‘indices’, ‘cluster’

get (*args, **kwargs)

Get a typed JSON document from the index based on its id.
<http://www.elastic.co/guide/en/elasticsearch/reference/current/docs-get.html>

Parameters

- **index** – The name of the index
- **doc_type** – The type of the document (use *_all* to fetch the first document matching the ID across all types)
- **id** – The document ID
- **_source** – True or false to return the *_source* field or not, or a list of fields to return
- **_source_exclude** – A list of fields to exclude from the returned *_source* field
- **_source_include** – A list of fields to extract and return from the *_source* field
- **fields** – A comma-separated list of fields to return in the response
- **parent** – The ID of the parent document
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **realtime** – Specify whether to perform the operation in realtime or search mode
- **refresh** – Refresh the shard containing the document before performing the operation
- **routing** – Specific routing value
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: ‘internal’, ‘external’, ‘external_gte’, ‘force’

get_script (*args, **kwargs)

Retrieve a script from the API. <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-scripting.html>

Parameters

- **lang** – Script language
- **id** – Script ID
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: ‘internal’, ‘external’, ‘external_gte’, ‘force’

get_source (*args, **kwargs)

Get the source of a document by its index, type and id.
<http://www.elastic.co/guide/en/elasticsearch/reference/current/docs-get.html>

Parameters

- **index** – The name of the index
- **doc_type** – The type of the document; use *_all* to fetch the first document matching the ID across all types
- **id** – The document ID
- **_source** – True or false to return the *_source* field or not, or a list of fields to return
- **_source_exclude** – A list of fields to exclude from the returned *_source* field
- **_source_include** – A list of fields to extract and return from the *_source* field
- **parent** – The ID of the parent document
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **realtime** – Specify whether to perform the operation in realtime or search mode
- **refresh** – Refresh the shard containing the document before performing the operation
- **routing** – Specific routing value
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: 'internal', 'external', 'external_gte', 'force'

get_template (*args, **kwargs)

Retrieve a search template. <http://www.elastic.co/guide/en/elasticsearch/reference/current/search-template.html>

Parameters

- **id** – Template ID
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: 'internal', 'external', 'external_gte', 'force'

index (*args, **kwargs)

Adds or updates a typed JSON document in a specific index, making it searchable. http://www.elastic.co/guide/en/elasticsearch/reference/current/docs-index_.html

Parameters

- **index** – The name of the index
- **doc_type** – The type of the document
- **body** – The document
- **id** – Document ID
- **consistency** – Explicit write consistency setting for the operation, valid choices are: 'one', 'quorum', 'all'
- **op_type** – Explicit operation type, default 'index', valid choices are: 'index', 'create'
- **parent** – ID of the parent document
- **refresh** – Refresh the index after performing the operation
- **routing** – Specific routing value
- **timeout** – Explicit operation timeout

- **timestamp** – Explicit timestamp for the document
- **t1** – Expiration time for the document
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: ‘internal’, ‘external’, ‘external_gte’, ‘force’

info (*args, **kwargs)

Get the basic info from the current cluster. <http://www.elastic.co/guide/>

mget (*args, **kwargs)

Get multiple documents based on an index, type (optional) and ids. <http://www.elastic.co/guide/en/elasticsearch/reference/current/docs-multi-get.html>

Parameters

- **body** – Document identifiers; can be either *docs* (containing full document information) or *ids* (when index and type is provided in the URL).
- **index** – The name of the index
- **doc_type** – The type of the document
- **_source** – True or false to return the *_source* field or not, or a list of fields to return
- **_source_exclude** – A list of fields to exclude from the returned *_source* field
- **_source_include** – A list of fields to extract and return from the *_source* field
- **fields** – A comma-separated list of fields to return in the response
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **realtime** – Specify whether to perform the operation in realtime or search mode
- **refresh** – Refresh the shard containing the document before performing the operation

mpercolate (*args, **kwargs)

The percolator allows to register queries against an index, and then send percolate requests which include a doc, and getting back the queries that match on that doc out of the set of registered queries. <http://www.elastic.co/guide/en/elasticsearch/reference/current/search-percolate.html>

Parameters

- **body** – The percolate request definitions (header & body pair), separated by newlines
- **index** – The index of the document being count percolated to use as default
- **doc_type** – The type of the document being percolated to use as default.
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

msearch (*args, **kwargs)

Execute several search requests within the same API. <http://www.elastic.co/guide/en/elasticsearch/reference/current/search-multi-search.html>

Parameters

- **body** – The request definitions (metadata-search request definition pairs), separated by newlines
- **index** – A comma-separated list of index names to use as default
- **doc_type** – A comma-separated list of document types to use as default
- **search_type** – Search operation type, valid choices are: ‘query_then_fetch’, ‘query_and_fetch’, ‘dfs_query_then_fetch’, ‘dfs_query_and_fetch’, ‘count’, ‘scan’

termvectors (*args, **kwargs)

Multi termvectors API allows to get multiple termvectors based on an index, type and id.
<http://www.elastic.co/guide/en/elasticsearch/reference/current/docs-multi-termvectors.html>

Parameters

- **index** – The index in which the document resides.
- **doc_type** – The type of the document.
- **body** – Define ids, documents, parameters or a list of parameters per document here. You must at least provide a list of document ids. See documentation.
- **field_statistics** – Specifies if document count, sum of document frequencies and sum of total term frequencies should be returned. Applies to all returned documents unless otherwise specified in body “params” or “docs”., default True
- **fields** – A comma-separated list of fields to return. Applies to all returned documents unless otherwise specified in body “params” or “docs”.
- **ids** – A comma-separated list of documents ids. You must define ids as parameter or set “ids” or “docs” in the request body
- **offsets** – Specifies if term offsets should be returned. Applies to all returned documents unless otherwise specified in body “params” or “docs”., default True
- **parent** – Parent id of documents. Applies to all returned documents unless otherwise specified in body “params” or “docs”.
- **payloads** – Specifies if term payloads should be returned. Applies to all returned documents unless otherwise specified in body “params” or “docs”., default True
- **positions** – Specifies if term positions should be returned. Applies to all returned documents unless otherwise specified in body “params” or “docs”., default True
- **preference** – Specify the node or shard the operation should be performed on (default: random) .Applies to all returned documents unless otherwise specified in body “params” or “docs”.
- **realtime** – Specifies if requests are real-time as opposed to near- real-time (default: true).
- **routing** – Specific routing value. Applies to all returned documents unless otherwise specified in body “params” or “docs”.
- **term_statistics** – Specifies if total term frequency and document frequency should be returned. Applies to all returned documents unless otherwise specified in body “params” or “docs”., default False
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: ‘internal’, ‘external’, ‘external_gte’, ‘force’

percolate (*args, **kwargs)

The percolator allows to register queries against an index, and then send percolate requests which include a doc, and getting back the queries that match on that doc out of the set of registered queries. <http://www.elastic.co/guide/en/elasticsearch/reference/current/search-percolate.html>

Parameters

- **index** – The index of the document being percolated.
- **doc_type** – The type of the document being percolated.
- **id** – Substitute the document in the request body with a document that is known by the specified id. On top of the id, the index and type parameter will be used to retrieve the document from within the cluster.
- **body** – The percolator request definition using the percolate DSL
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open', valid choices are: 'open', 'closed', 'none', 'all'
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **percolate_format** – Return an array of matching query IDs instead of objects, valid choices are: 'ids'
- **percolate_index** – The index to percolate the document into. Defaults to index.
- **percolate_preference** – Which shard to prefer when executing the percolate request.
- **percolate_routing** – The routing value to use when percolating the existing document.
- **percolate_type** – The type to percolate document into. Defaults to type.
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **routing** – A comma-separated list of specific routing values
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: 'internal', 'external', 'external_gte', 'force'

ping (*args, **kwargs)

Returns True if the cluster is up, False otherwise. <http://www.elastic.co/guide/>

put_script (*args, **kwargs)

Create a script in given language with specified ID. <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-scripting.html>

Parameters

- **lang** – Script language
- **id** – Script ID
- **body** – The document
- **op_type** – Explicit operation type, default 'index', valid choices are: 'index', 'create'

- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: ‘internal’, ‘external’, ‘external_gte’, ‘force’

put_template (*args, **kwargs)

Create a search template. <http://www.elastic.co/guide/en/elasticsearch/reference/current/search-template.html>

Parameters

- **id** – Template ID
- **body** – The document
- **op_type** – Explicit operation type, default ‘index’, valid choices are: ‘index’, ‘create’
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: ‘internal’, ‘external’, ‘external_gte’, ‘force’

render_search_template (*args, **kwargs)

<http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/search-template.html>

Parameters

- **id** – The id of the stored search template
- **body** – The search definition template and its params

scroll (*args, **kwargs)

Scroll a search request created by specifying the scroll parameter. <http://www.elastic.co/guide/en/elasticsearch/reference/current/search-request-scroll.html>

Parameters

- **scroll_id** – The scroll ID
- **body** – The scroll ID if not passed by URL or query parameter.
- **scroll** – Specify how long a consistent view of the index should be maintained for scrolled search

search (*args, **kwargs)

Execute a search query and get back search hits that match the query. <http://www.elastic.co/guide/en/elasticsearch/reference/current/search-search.html>

Parameters

- **index** – A comma-separated list of index names to search; use *_all* or empty string to perform the operation on all indices
- **doc_type** – A comma-separated list of document types to search; leave empty to perform the operation on all types
- **body** – The search definition using the Query DSL
- **_source** – True or false to return the *_source* field or not, or a list of fields to return
- **_source_exclude** – A list of fields to exclude from the returned *_source* field
- **_source_include** – A list of fields to extract and return from the *_source* field
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)

- **analyze_wildcard** – Specify whether wildcard and prefix queries should be analyzed (default: false)
- **analyzer** – The analyzer to use for the query string
- **default_operator** – The default operator for query string query (AND or OR), default 'OR', valid choices are: 'AND', 'OR'
- **df** – The field to use as default where no field prefix is given in the query string
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open', valid choices are: 'open', 'closed', 'none', 'all'
- **explain** – Specify whether to return detailed information about score computation as part of a hit
- **fielddata_fields** – A comma-separated list of fields to return as the field data representation of a field for each hit
- **fields** – A comma-separated list of fields to return as part of a hit
- **from** – Starting offset (default: 0)
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **lenient** – Specify whether format-based query failures (such as providing text to a numeric field) should be ignored
- **lowercase_expanded_terms** – Specify whether query terms should be lowercased
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **q** – Query in the Lucene query string syntax
- **request_cache** – Specify if request cache should be used for this request or not, defaults to index level setting
- **routing** – A comma-separated list of specific routing values
- **scroll** – Specify how long a consistent view of the index should be maintained for scrolled search
- **search_type** – Search operation type, valid choices are: 'query_then_fetch', 'dfs_query_then_fetch', 'count', 'scan'
- **size** – Number of hits to return (default: 10)
- **sort** – A comma-separated list of <field>:<direction> pairs
- **stats** – Specific 'tag' of the request for logging and statistical purposes
- **suggest_field** – Specify which field to use for suggestions
- **suggest_mode** – Specify suggest mode, default 'missing', valid choices are: 'missing', 'popular', 'always'
- **suggest_size** – How many suggestions to return in response
- **suggest_text** – The source text for which the suggestions should be returned
- **terminate_after** – The maximum number of documents to collect for each shard, upon reaching which the query execution will terminate early.
- **timeout** – Explicit operation timeout

- **track_scores** – Whether to calculate and return scores even if they are not used for sorting
- **version** – Specify whether to return document version as part of a hit

search_exists (*args, **kwargs)

The exists API allows to easily determine if any matching documents exist for a provided query.

<http://www.elastic.co/guide/en/elasticsearch/reference/current/search-exists.html>

Parameters

- **index** – A comma-separated list of indices to restrict the results
- **doc_type** – A comma-separated list of types to restrict the results
- **body** – A query to restrict the results specified with the Query DSL (optional)
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **analyze_wildcard** – Specify whether wildcard and prefix queries should be analyzed (default: false)
- **analyzer** – The analyzer to use for the query string
- **default_operator** – The default operator for query string query (AND or OR), default 'OR', valid choices are: 'AND', 'OR'
- **df** – The field to use as default where no field prefix is given in the query string
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open', valid choices are: 'open', 'closed', 'none', 'all'
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **lenient** – Specify whether format-based query failures (such as providing text to a numeric field) should be ignored
- **lowercase_expanded_terms** – Specify whether query terms should be lowercased
- **min_score** – Include only documents with a specific *_score* value in the result
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **q** – Query in the Lucene query string syntax
- **routing** – Specific routing value

search_shards (*args, **kwargs)

The search shards api returns the indices and shards that a search request would be executed against. This can give useful feedback for working out issues or planning optimizations with routing and shard preferences. <http://www.elastic.co/guide/en/elasticsearch/reference/current/search-shards.html>

Parameters

- **index** – A comma-separated list of index names to search; use *_all* or empty string to perform the operation on all indices
- **doc_type** – A comma-separated list of document types to search; leave empty to perform the operation on all types
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)

- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **routing** – Specific routing value

search_template (*args, **kwargs)

A query that accepts a query template and a map of key/value pairs to fill in template parameters.
<http://www.elastic.co/guide/en/elasticsearch/reference/current/search-template.html>

Parameters

- **index** – A comma-separated list of index names to search; use *_all* or empty string to perform the operation on all indices
- **doc_type** – A comma-separated list of document types to search; leave empty to perform the operation on all types
- **body** – The search definition template and its params
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **routing** – A comma-separated list of specific routing values
- **scroll** – Specify how long a consistent view of the index should be maintained for scrolled search
- **search_type** – Search operation type, valid choices are: ‘query_then_fetch’, ‘query_and_fetch’, ‘dfs_query_then_fetch’, ‘dfs_query_and_fetch’, ‘count’, ‘scan’

suggest (*args, **kwargs)

The suggest feature suggests similar looking terms based on a provided text by using a suggester.
<http://www.elastic.co/guide/en/elasticsearch/reference/current/search-suggesters.html>

Parameters

- **body** – The request definition
- **index** – A comma-separated list of index names to restrict the operation; use *_all* or empty string to perform the operation on all indices
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’

- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **routing** – Specific routing value

termvectors (*args, **kwargs)

Returns information and statistics on terms in the fields of a particular document. The document could be stored in the index or artificially provided by the user (Added in 1.4). Note that for documents stored in the index, this is a near realtime API as the term vectors are not available until the next refresh. <http://www.elastic.co/guide/en/elasticsearch/reference/current/docs-termvectors.html>

Parameters

- **index** – The index in which the document resides.
- **doc_type** – The type of the document.
- **id** – The id of the document, when not specified a doc param should be supplied.
- **body** – Define parameters and or supply a document to get termvectors for. See documentation.
- **dfs** – Specifies if distributed frequencies should be returned instead shard frequencies., default False
- **field_statistics** – Specifies if document count, sum of document frequencies and sum of total term frequencies should be returned., default True
- **fields** – A comma-separated list of fields to return.
- **offsets** – Specifies if term offsets should be returned., default True
- **parent** – Parent id of documents.
- **payloads** – Specifies if term payloads should be returned., default True
- **positions** – Specifies if term positions should be returned., default True
- **preference** – Specify the node or shard the operation should be performed on (default: random).
- **realtime** – Specifies if request is real-time as opposed to near- real-time (default: true).
- **routing** – Specific routing value.
- **term_statistics** – Specifies if total term frequency and document frequency should be returned., default False
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: 'internal', 'external', 'external_gte', 'force'

update (*args, **kwargs)

Update a document based on a script or partial data provided. <http://www.elastic.co/guide/en/elasticsearch/reference/current/docs-update.html>

Parameters

- **index** – The name of the index
- **doc_type** – The type of the document
- **id** – Document ID

- **body** – The request definition using either *script* or partial *doc*
- **consistency** – Explicit write consistency setting for the operation, valid choices are: ‘one’, ‘quorum’, ‘all’
- **detect_noop** – Specifying as true will cause Elasticsearch to check if there are changes and, if there aren’t, turn the update request into a noop.
- **fields** – A comma-separated list of fields to return in the response
- **lang** – The script language (default: groovy)
- **parent** – ID of the parent document. Is only used for routing and when for the upsert request
- **refresh** – Refresh the index after performing the operation
- **retry_on_conflict** – Specify how many times should the operation be retried when a conflict occurs (default: 0)
- **routing** – Specific routing value
- **script** – The URL-encoded script definition (instead of using request body)
- **script_id** – The id of a stored script
- **scripted_upsert** – True if the script referenced in script or script_id should be called to perform inserts - defaults to false
- **timeout** – Explicit operation timeout
- **timestamp** – Explicit timestamp for the document
- **t1** – Expiration time for the document
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type, valid choices are: ‘internal’, ‘force’

5.1.3 Indices

`class elasticsearch.client.IndicesClient (client)`

analyze (*args, **kwargs)

Perform the analysis process on a text and return the tokens breakdown of the text.
<http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-analyze.html>

Parameters

- **index** – The name of the index to scope the operation
- **body** – The text on which the analysis should be performed
- **analyzer** – The name of the analyzer to use
- **char_filters** – A comma-separated list of character filters to use for the analysis
- **field** – Use the analyzer configured for this field (instead of passing the analyzer name)
- **filters** – A comma-separated list of filters to use for the analysis
- **format** – Format of the output, default ‘detailed’, valid choices are: ‘detailed’, ‘text’
- **prefer_local** – With *true*, specify that a local shard should be used if available, with *false*, use a random shard (default: true)

- **text** – The text on which the analysis should be performed (when request body is not used)
- **tokenizer** – The name of the tokenizer to use for the analysis

clear_cache (*args, **kwargs)

Clear either all caches or specific cached associated with one ore more indices.
<http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-clearcache.html>

Parameters

- **index** – A comma-separated list of index name to limit the operation
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open', valid choices are: 'open', 'closed', 'none', 'all'
- **field_data** – Clear field data
- **fielddata** – Clear field data
- **fields** – A comma-separated list of fields to clear when using the *field_data* parameter (default: all)
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **query** – Clear query caches
- **recycler** – Clear the recycler cache
- **request** – Clear request cache

close (*args, **kwargs)

Close an index to remove it's overhead from the cluster. Closed index is blocked for read/write operations.
<http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-open-close.html>

Parameters

- **index** – The name of the index
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open', valid choices are: 'open', 'closed', 'none', 'all'
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit operation timeout

create (*args, **kwargs)

Create an index in Elasticsearch. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-create-index.html>

Parameters

- **index** – The name of the index
- **body** – The configuration for the index (*settings* and *mappings*)
- **master_timeout** – Specify timeout for connection to master

- **timeout** – Explicit operation timeout
- **update_all_types** – Whether to update the mapping for all fields with the same name across all types or not

delete (*args, **kwargs)

Delete an index in Elasticsearch <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-delete-index.html>

Parameters

- **index** – A comma-separated list of indices to delete; use *_all* or *** string to delete all indices
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit operation timeout

delete_alias (*args, **kwargs)

Delete specific alias. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-aliases.html>

Parameters

- **index** – A comma-separated list of index names (supports wildcards); use *_all* for all indices
- **name** – A comma-separated list of aliases to delete (supports wildcards); use *_all* to delete all aliases for the specified indices.
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit timestamp for the document

delete_template (*args, **kwargs)

Delete an index template by its name. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-templates.html>

Parameters

- **name** – The name of the template
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit operation timeout

delete_warmer (*args, **kwargs)

Delete an index warmer. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-warmers.html>

Parameters

- **index** – A comma-separated list of index names to delete warmers from (supports wildcards); use *_all* to perform the operation on all indices.
- **name** – A comma-separated list of warmer names to delete (supports wildcards); use *_all* to delete all warmers in the specified indices. You must specify a name either in the uri or in the parameters.
- **master_timeout** – Specify timeout for connection to master

exists (*args, **kwargs)

Return a boolean indicating whether given index exists. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-exists.html>

Parameters

- **index** – A comma-separated list of indices to check
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open', valid choices are: 'open', 'closed', 'none', 'all'
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **local** – Return local information, do not retrieve the state from master node (default: false)

exists_alias (*args, **kwargs)

Return a boolean indicating whether given alias exists. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-aliases.html>

Parameters

- **index** – A comma-separated list of index names to filter aliases
- **name** – A comma-separated list of alias names to return
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ['open', 'closed'], valid choices are: 'open', 'closed', 'none', 'all'
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **local** – Return local information, do not retrieve the state from master node (default: false)

exists_template (*args, **kwargs)

Return a boolean indicating whether given template exists. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-templates.html>

Parameters

- **name** – The name of the template
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node

exists_type (*args, **kwargs)

Check if a type/types exists in an index/indices. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-types-exists.html>

Parameters

- **index** – A comma-separated list of index names; use *_all* to check the types across all indices
- **doc_type** – A comma-separated list of document types to check
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)

- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **local** – Return local information, do not retrieve the state from master node (default: false)

flush (*args, **kwargs)

Explicitly flush one or more indices. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-flush.html>

Parameters

- **index** – A comma-separated list of index names; use *_all* or empty string for all indices
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **force** – Whether a flush should be forced even if it is not necessarily needed ie. if no changes will be committed to the index. This is useful if transaction log IDs should be incremented even if no uncommitted changes are present. (This setting can be considered as internal)
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **wait_if_ongoing** – If set to true the flush operation will block until the flush can be executed if another flush operation is already executing. The default is false and will cause an exception to be thrown on the shard level if another flush operation is already running.

flush_synced (*args, **kwargs)

Perform a normal flush, then add a generated unique marker (sync_id) to all shards. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-synced-flush.html>

Parameters

- **index** – A comma-separated list of index names; use *_all* or empty string for all indices
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

get (*args, **kwargs)

The get index API allows to retrieve information about one or more indexes. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-get-index.html>

Parameters

- **index** – A comma-separated list of index names
- **feature** – A comma-separated list of features
- **allow_no_indices** – Ignore if a wildcard expression resolves to no concrete indices (default: false)

- **expand_wildcards** – Whether wildcard expressions should get expanded to open or closed indices (default: open), default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **flat_settings** – Return settings in flat format (default: false)
- **human** – Whether to return version and creation date values in human- readable format., default False
- **ignore_unavailable** – Ignore unavailable indexes (default: false)
- **local** – Return local information, do not retrieve the state from master node (default: false)

get_alias (*args, **kwargs)

Retrieve a specified alias. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-aliases.html>

Parameters

- **index** – A comma-separated list of index names to filter aliases
- **name** – A comma-separated list of alias names to return
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **local** – Return local information, do not retrieve the state from master node (default: false)

get_aliases (*args, **kwargs)

Retrieve specified aliases <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-aliases.html>

Parameters

- **index** – A comma-separated list of index names to filter aliases
- **name** – A comma-separated list of alias names to filter
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **timeout** – Explicit operation timeout

get_field_mapping (*args, **kwargs)

Retrieve mapping definition of a specific field. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-get-field-mapping.html>

Parameters

- **fields** – A comma-separated list of fields
- **index** – A comma-separated list of index names
- **doc_type** – A comma-separated list of document types
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)

- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **include_defaults** – Whether the default mapping values should be returned as well
- **local** – Return local information, do not retrieve the state from master node (default: false)

get_mapping (*args, **kwargs)

Retrieve mapping definition of index or index/type. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-get-mapping.html>

Parameters

- **index** – A comma-separated list of index names
- **doc_type** – A comma-separated list of document types
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **local** – Return local information, do not retrieve the state from master node (default: false)

get_settings (*args, **kwargs)

Retrieve settings for one or more (or all) indices. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-get-settings.html>

Parameters

- **index** – A comma-separated list of index names; use *_all* or empty string to perform the operation on all indices
- **name** – The name of the settings that should be included
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default [‘open’, ‘closed’], valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **flat_settings** – Return settings in flat format (default: false)
- **human** – Whether to return version and creation date values in human- readable format., default False
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **local** – Return local information, do not retrieve the state from master node (default: false)

get_template (*args, **kwargs)

Retrieve an index template by its name. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-templates.html>

Parameters

- **name** – The name of the template
- **flat_settings** – Return settings in flat format (default: false)
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node

get_upgrade (*args, **kwargs)

Monitor how much of one or more index is upgraded. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-upgrade.html>

Parameters

- **index** – A comma-separated list of index names; use *_all* or empty string to perform the operation on all indices
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open', valid choices are: 'open', 'closed', 'none', 'all'
- **human** – Whether to return time and byte values in human-readable format., default False
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

get_warmer (*args, **kwargs)

Retrieve an index warmer. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-warmers.html>

Parameters

- **index** – A comma-separated list of index names to restrict the operation; use *_all* to perform the operation on all indices
- **doc_type** – A comma-separated list of document types to restrict the operation; leave empty to perform the operation on all types
- **name** – The name of the warmer (supports wildcards); leave empty to get all warmers
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open', valid choices are: 'open', 'closed', 'none', 'all'
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **local** – Return local information, do not retrieve the state from master node (default: false)

open (*args, **kwargs)

Open a closed index to make it available for search. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-open-close.html>

Parameters

- **index** – The name of the index

- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘closed’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit operation timeout

optimize (*args, **kwargs)

Explicitly optimize one or more indices through an API. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-optimize.html>

Parameters

- **index** – A comma-separated list of index names; use *_all* or empty string to perform the operation on all indices
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **flush** – Specify whether the index should be flushed after performing the operation (default: true)
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **max_num_segments** – The number of segments the index should be merged into (default: dynamic)
- **only_expunge_deletes** – Specify whether the operation should only expunge deleted documents
- **operation_threading** – TODO: ?
- **wait_for_merge** – Specify whether the request should block until the merge process is finished (default: true)

put_alias (*args, **kwargs)

Create an alias for a specific index/indices. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-aliases.html>

Parameters

- **index** – A comma-separated list of index names the alias should point to (supports wildcards); use *_all* to perform the operation on all indices.
- **name** – The name of the alias to be created or updated
- **body** – The settings for the alias, such as *routing* or *filter*
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit timestamp for the document

put_mapping (*args, **kwargs)

Register specific mapping definition for a specific type. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-put-mapping.html>

Parameters

- **doc_type** – The name of the document type
- **body** – The mapping definition
- **index** – A comma-separated list of index names the mapping should be added to (supports wildcards); use *_all* or omit to add the mapping on all indices.
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open', valid choices are: 'open', 'closed', 'none', 'all'
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit operation timeout
- **update_all_types** – Whether to update the mapping for all fields with the same name across all types or not

put_settings (*args, **kwargs)

Change specific index level settings in real time. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-update-settings.html>

Parameters

- **body** – The index settings to be updated
- **index** – A comma-separated list of index names; use *_all* or empty string to perform the operation on all indices
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open', valid choices are: 'open', 'closed', 'none', 'all'
- **flat_settings** – Return settings in flat format (default: false)
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **master_timeout** – Specify timeout for connection to master

put_template (*args, **kwargs)

Create an index template that will automatically be applied to new indices created. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-templates.html>

Parameters

- **name** – The name of the template
- **body** – The template definition
- **create** – Whether the index template should only be added if new or can also replace an existing one, default False
- **flat_settings** – Return settings in flat format (default: false)
- **master_timeout** – Specify timeout for connection to master

- **order** – The order for this template when merging multiple matching ones (higher numbers are merged later, overriding the lower numbers)
- **timeout** – Explicit operation timeout

put_warmer (*args, **kwargs)

Create an index warmer to run registered search requests to warm up the index before it is available for search. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-warmers.html>

Parameters

- **name** – The name of the warmer
- **body** – The search request definition for the warmer (query, filters, facets, sorting, etc)
- **index** – A comma-separated list of index names to register the warmer for; use *_all* or omit to perform the operation on all indices
- **doc_type** – A comma-separated list of document types to register the warmer for; leave empty to perform the operation on all types
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices in the search request to warm. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both, in the search request to warm., default 'open', valid choices are: 'open', 'closed', 'none', 'all'
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed) in the search request to warm
- **master_timeout** – Specify timeout for connection to master
- **request_cache** – Specify whether the request to be warmed should use the request cache, defaults to index level setting

recovery (*args, **kwargs)

The indices recovery API provides insight into on-going shard recoveries. Recovery status may be reported for specific indices, or cluster-wide. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-recovery.html>

Parameters

- **index** – A comma-separated list of index names; use *_all* or empty string to perform the operation on all indices
- **active_only** – Display only those recoveries that are currently on- going, default False
- **detailed** – Whether to display detailed information about shard recovery, default False
- **human** – Whether to return time and byte values in human-readable format., default False

refresh (*args, **kwargs)

Explicitly refresh one or more index, making all operations performed since the last refresh available for search. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-refresh.html>

Parameters

- **index** – A comma-separated list of index names; use *_all* or empty string to perform the operation on all indices
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)

- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **force** – Force a refresh even if not required, default False
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **operation_threading** – TODO: ?

segments (*args, **kwargs)

Provide low level segments information that a Lucene index (shard level) is built with.
<http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-segments.html>

Parameters

- **index** – A comma-separated list of index names; use *_all* or empty string to perform the operation on all indices
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **human** – Whether to return time and byte values in human-readable format., default False
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **operation_threading** – TODO: ?

shard_stores (*args, **kwargs)

<http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-shard-stores.html>

Parameters

- **index** – A comma-separated list of index names; use *_all* or empty string to perform the operation on all indices
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **operation_threading** – TODO: ?
- **status** – A comma-separated list of statuses used to filter on shards to get store information for, valid choices are: ‘green’, ‘yellow’, ‘red’, ‘all’

stats (*args, **kwargs)

Retrieve statistics on different operations happening on an index.
<http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-stats.html>

Parameters

- **index** – A comma-separated list of index names; use *_all* or empty string to perform the operation on all indices
- **metric** – Limit the information returned the specific metrics.

- **completion_fields** – A comma-separated list of fields for *fielddata* and *suggest* index metric (supports wildcards)
- **fielddata_fields** – A comma-separated list of fields for *fielddata* index metric (supports wildcards)
- **fields** – A comma-separated list of fields for *fielddata* and *completion* index metric (supports wildcards)
- **groups** – A comma-separated list of search groups for *search* index metric
- **human** – Whether to return time and byte values in human-readable format., default False
- **level** – Return stats aggregated at cluster, index or shard level, default ‘indices’, valid choices are: ‘cluster’, ‘indices’, ‘shards’
- **types** – A comma-separated list of document types for the *indexing* index metric

update_aliases (*args, **kwargs)

Update specified aliases. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-aliases.html>

Parameters

- **body** – The definition of *actions* to perform
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Request timeout

upgrade (*args, **kwargs)

Upgrade one or more indices to the latest format through an API. <http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-upgrade.html>

Parameters

- **index** – A comma-separated list of index names; use *_all* or empty string to perform the operation on all indices
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’, valid choices are: ‘open’, ‘closed’, ‘none’, ‘all’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **only_ancient_segments** – If true, only ancient (an older Lucene major release) segments will be upgraded
- **wait_for_completion** – Specify whether the request should block until the all segments are upgraded (default: false)

validate_query (*args, **kwargs)

Validate a potentially expensive query without executing it. <http://www.elastic.co/guide/en/elasticsearch/reference/current/validate.html>

Parameters

- **index** – A comma-separated list of index names to restrict the operation; use *_all* or empty string to perform the operation on all indices
- **doc_type** – A comma-separated list of document types to restrict the operation; leave empty to perform the operation on all types

- **body** – The query definition specified with the Query DSL
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes *_all* string or when no indices have been specified)
- **analyze_wildcard** – Specify whether wildcard and prefix queries should be analyzed (default: false)
- **analyzer** – The analyzer to use for the query string
- **default_operator** – The default operator for query string query (AND or OR), default 'OR', valid choices are: 'AND', 'OR'
- **df** – The field to use as default where no field prefix is given in the query string
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open', valid choices are: 'open', 'closed', 'none', 'all'
- **explain** – Return detailed information about the error
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **lenient** – Specify whether format-based query failures (such as providing text to a numeric field) should be ignored
- **lowercase_expanded_terms** – Specify whether query terms should be lowercased
- **operation_threading** – TODO: ?
- **q** – Query in the Lucene query string syntax
- **rewrite** – Provide a more detailed explanation showing the actual Lucene query that will be executed.

5.1.4 Cluster

`class elasticsearch.client.ClusterClient` (*client*)

get_settings (**args, **kwargs*)

Get cluster settings. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-update-settings.html>

Parameters

- **flat_settings** – Return settings in flat format (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **timeout** – Explicit operation timeout

health (**args, **kwargs*)

Get a very simple status on the health of the cluster. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-health.html>

Parameters

- **index** – Limit the information returned to a specific index
- **level** – Specify the level of detail for returned information, default 'cluster', valid choices are: 'cluster', 'indices', 'shards'
- **local** – Return local information, do not retrieve the state from master node (default: false)

- **master_timeout** – Explicit operation timeout for connection to master node
- **timeout** – Explicit operation timeout
- **wait_for_active_shards** – Wait until the specified number of shards is active
- **wait_for_nodes** – Wait until the specified number of nodes is available
- **wait_for_relocating_shards** – Wait until the specified number of relocating shards is finished
- **wait_for_status** – Wait until cluster is in a specific state, default None, valid choices are: 'green', 'yellow', 'red'

pending_tasks (*args, **kwargs)

The pending cluster tasks API returns a list of any cluster-level changes (e.g. create index, update mapping, allocate or fail shard) which have not yet been executed. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-pending.html>

Parameters

- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Specify timeout for connection to master

put_settings (*args, **kwargs)

Update cluster wide specific settings. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-update-settings.html>

Parameters

- **body** – The settings to be updated. Can be either *transient* or *persistent* (survives cluster restart).
- **flat_settings** – Return settings in flat format (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **timeout** – Explicit operation timeout

reroute (*args, **kwargs)

Explicitly execute a cluster reroute allocation command including specific commands. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-reroute.html>

Parameters

- **body** – The definition of *commands* to perform (*move*, *cancel*, *allocate*)
- **dry_run** – Simulate the operation only and return the resulting state
- **explain** – Return an explanation of why the commands can or cannot be executed
- **master_timeout** – Explicit operation timeout for connection to master node
- **metric** – Limit the information returned to the specified metrics. Defaults to all but metadata, valid choices are: '_all', 'blocks', 'metadata', 'nodes', 'routing_table', 'master_node', 'version'
- **timeout** – Explicit operation timeout

state (*args, **kwargs)

Get a comprehensive state information of the whole cluster. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-state.html>

Parameters

- **metric** – Limit the information returned to the specified metrics
- **index** – A comma-separated list of index names; use `_all` or empty string to perform the operation on all indices
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes `_all` string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open', valid choices are: 'open', 'closed', 'none', 'all'
- **flat_settings** – Return settings in flat format (default: false)
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Specify timeout for connection to master

stats (*args, **kwargs)

The Cluster Stats API allows to retrieve statistics from a cluster wide perspective. The API returns basic index metrics and information about the current nodes that form the cluster. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-stats.html>

Parameters

- **node_id** – A comma-separated list of node IDs or names to limit the returned information; use `_local` to return information from the node you're connecting to, leave empty to get information from all nodes
- **flat_settings** – Return settings in flat format (default: false)
- **human** – Whether to return time and byte values in human-readable format., default False
- **timeout** – Explicit operation timeout

5.1.5 Nodes

class `elasticsearch.client.NodesClient` (*client*)

hot_threads (*args, **kwargs)

An API allowing to get the current hot threads on each node in the cluster. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-nodes-hot-threads.html>

Parameters

- **node_id** – A comma-separated list of node IDs or names to limit the returned information; use `_local` to return information from the node you're connecting to, leave empty to get information from all nodes
- **doc_type** – The type to sample (default: cpu), valid choices are: 'cpu', 'wait', 'block'
- **ignore_idle_threads** – Don't show threads that are in known-idle places, such as waiting on a socket select or pulling from an empty task queue (default: true)
- **interval** – The interval for the second sampling of threads
- **snapshots** – Number of samples of thread stacktrace (default: 10)
- **threads** – Specify the number of threads to provide information for (default: 3)

- **timeout** – Explicit operation timeout

info (*args, **kwargs)

The cluster nodes info API allows to retrieve one or more (or all) of the cluster nodes information.

<http://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-nodes-info.html>

Parameters

- **node_id** – A comma-separated list of node IDs or names to limit the returned information; use *_local* to return information from the node you’re connecting to, leave empty to get information from all nodes
- **metric** – A comma-separated list of metrics you wish returned. Leave empty to return all.
- **flat_settings** – Return settings in flat format (default: false)
- **human** – Whether to return time and byte values in human-readable format., default False
- **timeout** – Explicit operation timeout

stats (*args, **kwargs)

The cluster nodes stats API allows to retrieve one or more (or all) of the cluster nodes statistics.

<http://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-nodes-stats.html>

Parameters

- **node_id** – A comma-separated list of node IDs or names to limit the returned information; use *_local* to return information from the node you’re connecting to, leave empty to get information from all nodes
- **metric** – Limit the information returned to the specified metrics
- **index_metric** – Limit the information returned for *indices* metric to the specific index metrics. Isn’t used if *indices* (or *all*) metric isn’t specified.
- **completion_fields** – A comma-separated list of fields for *fielddata* and *suggest* index metric (supports wildcards)
- **fielddata_fields** – A comma-separated list of fields for *fielddata* index metric (supports wildcards)
- **fields** – A comma-separated list of fields for *fielddata* and *completion* index metric (supports wildcards)
- **groups** – A comma-separated list of search groups for *search* index metric
- **human** – Whether to return time and byte values in human-readable format., default False
- **level** – Return indices stats aggregated at node, index or shard level, default ‘node’, valid choices are: ‘node’, ‘indices’, ‘shards’
- **timeout** – Explicit operation timeout
- **types** – A comma-separated list of document types for the *indexing* index metric

5.1.6 Cat

class `elasticsearch.client.CatClient` (*client*)

aliases (*args, **kwargs)

<http://www.elastic.co/guide/en/elasticsearch/reference/current/cat-alias.html>

Parameters

- **name** – A comma-separated list of alias names to return
- **h** – Comma-separated list of column names to display
- **help** – Return help information, default False
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **v** – Verbose mode. Display column headers, default False

allocation (*args, **kwargs)

Allocation provides a snapshot of how shards have located around the cluster and the state of disk usage.
<http://www.elastic.co/guide/en/elasticsearch/reference/current/cat-allocation.html>

Parameters

- **node_id** – A comma-separated list of node IDs or names to limit the returned information
- **bytes** – The unit in which to display byte values, valid choices are: ‘b’, ‘k’, ‘m’, ‘g’
- **h** – Comma-separated list of column names to display
- **help** – Return help information, default False
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **v** – Verbose mode. Display column headers, default False

count (*args, **kwargs)

Count provides quick access to the document count of the entire cluster, or individual indices.
<http://www.elastic.co/guide/en/elasticsearch/reference/current/cat-count.html>

Parameters

- **index** – A comma-separated list of index names to limit the returned information
- **h** – Comma-separated list of column names to display
- **help** – Return help information, default False
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **v** – Verbose mode. Display column headers, default False

fielddata (*args, **kwargs)

Shows information about currently loaded fielddata on a per-node basis.
<http://www.elastic.co/guide/en/elasticsearch/reference/current/cat-fielddata.html>

Parameters

- **fields** – A comma-separated list of fields to return the fielddata size
- **bytes** – The unit in which to display byte values, valid choices are: ‘b’, ‘k’, ‘m’, ‘g’
- **h** – Comma-separated list of column names to display

- **help** – Return help information, default False
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **v** – Verbose mode. Display column headers, default False

health (*args, **kwargs)

health is a terse, one-line representation of the same information from `health()` API <http://www.elastic.co/guide/en/elasticsearch/reference/current/cat-health.html>

Parameters

- **h** – Comma-separated list of column names to display
- **help** – Return help information, default False
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **ts** – Set to false to disable timestamping, default True
- **v** – Verbose mode. Display column headers, default False

help (*args, **kwargs)

A simple help for the cat api. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cat.html>

Parameters **help** – Return help information, default False

indices (*args, **kwargs)

The indices command provides a cross-section of each index. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cat-indices.html>

Parameters

- **index** – A comma-separated list of index names to limit the returned information
- **bytes** – The unit in which to display byte values, valid choices are: ‘b’, ‘k’, ‘m’, ‘g’
- **h** – Comma-separated list of column names to display
- **help** – Return help information, default False
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **pri** – Set to true to return stats only for primary shards, default False
- **v** – Verbose mode. Display column headers, default False

master (*args, **kwargs)

Displays the master’s node ID, bound IP address, and node name. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cat-master.html>

Parameters

- **h** – Comma-separated list of column names to display
- **help** – Return help information, default False
- **local** – Return local information, do not retrieve the state from master node (default: false)

- **master_timeout** – Explicit operation timeout for connection to master node
- **v** – Verbose mode. Display column headers, default False

nodeattrs (*args, **kwargs)

<http://www.elastic.co/guide/en/elasticsearch/reference/current/cat-nodeattrs.html>

Parameters

- **h** – Comma-separated list of column names to display
- **help** – Return help information, default False
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **v** – Verbose mode. Display column headers, default False

nodes (*args, **kwargs)

The nodes command shows the cluster topology. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cat-nodes.html>

Parameters

- **h** – Comma-separated list of column names to display
- **help** – Return help information, default False
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **v** – Verbose mode. Display column headers, default False

pending_tasks (*args, **kwargs)

pending_tasks provides the same information as the `pending_tasks()` API in a convenient tabular format. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cat-pending-tasks.html>

Parameters

- **h** – Comma-separated list of column names to display
- **help** – Return help information, default False
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **v** – Verbose mode. Display column headers, default False

plugins (*args, **kwargs)

<http://www.elastic.co/guide/en/elasticsearch/reference/current/cat-plugins.html>

Parameters

- **h** – Comma-separated list of column names to display
- **help** – Return help information, default False
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node

- **v** – Verbose mode. Display column headers, default False

recovery (*args, **kwargs)

recovery is a view of shard replication. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cat-recovery.html>

Parameters

- **index** – A comma-separated list of index names to limit the returned information
- **bytes** – The unit in which to display byte values, valid choices are: ‘b’, ‘k’, ‘m’, ‘g’
- **h** – Comma-separated list of column names to display
- **help** – Return help information, default False
- **master_timeout** – Explicit operation timeout for connection to master node
- **v** – Verbose mode. Display column headers, default False

segments (*args, **kwargs)

The segments command is the detailed view of Lucene segments per index. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cat-segments.html>

Parameters

- **index** – A comma-separated list of index names to limit the returned information
- **h** – Comma-separated list of column names to display
- **help** – Return help information, default False
- **v** – Verbose mode. Display column headers, default False

shards (*args, **kwargs)

The shards command is the detailed view of what nodes contain which shards. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cat-shards.html>

Parameters

- **index** – A comma-separated list of index names to limit the returned information
- **h** – Comma-separated list of column names to display
- **help** – Return help information, default False
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **v** – Verbose mode. Display column headers, default False

thread_pool (*args, **kwargs)

Get information about thread pools. <http://www.elastic.co/guide/en/elasticsearch/reference/current/cat-thread-pool.html>

Parameters

- **full_id** – Enables displaying the complete node ids, default False
- **h** – Comma-separated list of column names to display
- **help** – Return help information, default False
- **local** – Return local information, do not retrieve the state from master node (default: false)

- **master_timeout** – Explicit operation timeout for connection to master node
- **v** – Verbose mode. Display column headers, default False

Snapshot —

class `elasticsearch.client.SnapshotClient` (*client*)

create (**args, **kwargs*)

Create a snapshot in repository <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-snapshots.html>

Parameters

- **repository** – A repository name
- **snapshot** – A snapshot name
- **body** – The snapshot definition
- **master_timeout** – Explicit operation timeout for connection to master node
- **wait_for_completion** – Should this request wait until the operation has completed before returning, default False

create_repository (**args, **kwargs*)

Registers a shared file system repository. <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-snapshots.html>

Parameters

- **repository** – A repository name
- **body** – The repository definition
- **master_timeout** – Explicit operation timeout for connection to master node
- **timeout** – Explicit operation timeout
- **verify** – Whether to verify the repository after creation

delete (**args, **kwargs*)

Deletes a snapshot from a repository. <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-snapshots.html>

Parameters

- **repository** – A repository name
- **snapshot** – A snapshot name
- **master_timeout** – Explicit operation timeout for connection to master node

delete_repository (**args, **kwargs*)

Removes a shared file system repository. <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-snapshots.html>

Parameters

- **repository** – A comma-separated list of repository names
- **master_timeout** – Explicit operation timeout for connection to master node
- **timeout** – Explicit operation timeout

get (*args, **kwargs)

Retrieve information about a snapshot. <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-snapshots.html>

Parameters

- **repository** – A repository name
- **snapshot** – A comma-separated list of snapshot names
- **master_timeout** – Explicit operation timeout for connection to master node

get_repository (*args, **kwargs)

Return information about registered repositories. <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-snapshots.html>

Parameters

- **repository** – A comma-separated list of repository names
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node

restore (*args, **kwargs)

Restore a snapshot. <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-snapshots.html>

Parameters

- **repository** – A repository name
- **snapshot** – A snapshot name
- **body** – Details of what to restore
- **master_timeout** – Explicit operation timeout for connection to master node
- **wait_for_completion** – Should this request wait until the operation has completed before returning, default False

status (*args, **kwargs)

Return information about all currently running snapshots. By specifying a repository name, it's possible to limit the results to a particular repository. <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-snapshots.html>

Parameters

- **repository** – A repository name
- **snapshot** – A comma-separated list of snapshot names
- **master_timeout** – Explicit operation timeout for connection to master node

verify_repository (*args, **kwargs)

Returns a list of nodes where repository was successfully verified or an error message if verification process failed. <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-snapshots.html>

Parameters

- **repository** – A repository name
- **master_timeout** – Explicit operation timeout for connection to master node
- **timeout** – Explicit operation timeout

5.2 Exceptions

```
class elasticsearch.ImproperlyConfigured
    Exception raised when the config passed to the client is inconsistent or invalid.

class elasticsearch.ElasticsearchException
    Base class for all exceptions raised by this package's operations (doesn't apply to
    ImproperlyConfigured).

class elasticsearch.SerializationError (ElasticsearchException)
    Data passed in failed to serialize properly in the Serializer being used.

class elasticsearch.TransportError (ElasticsearchException)
    Exception raised when ES returns a non-OK (>=400) HTTP status code. Or when an actual connection error
    happens; in that case the status_code will be set to 'N/A'.

    error
        A string error message.

    info
        Dict of returned error info from ES, where available.

    status_code
        The HTTP status code of the response that precipitated the error or 'N/A' if not applicable.

class elasticsearch.ConnectionError (TransportError)
    Error raised when there was an exception while talking to ES. Original exception from the underlying
    Connection implementation is available as .info.

class elasticsearch.ConnectionTimeout (ConnectionError)
    A network timeout. Doesn't cause a node retry by default.

class elasticsearch.SSLError (ConnectionError)
    Error raised when encountering SSL errors.

class elasticsearch.NotFoundError (TransportError)
    Exception representing a 404 status code.

class elasticsearch.ConflictError (TransportError)
    Exception representing a 409 status code.

class elasticsearch.RequestError (TransportError)
    Exception representing a 400 status code.

class elasticsearch.ConnectionError (TransportError)
    Error raised when there was an exception while talking to ES. Original exception from the underlying
    Connection implementation is available as .info.
```

5.3 Connection Layer API

All of the classes responsible for handling the connection to the Elasticsearch cluster. The default subclasses used can be overridden by passing parameters to the *Elasticsearch* class. All of the arguments to the client will be passed on to *Transport*, *ConnectionPool* and *Connection*.

For example if you wanted to use your own implementation of the *ConnectionSelector* class you can just pass in the `selector_class` parameter.

Note: *ConnectionPool* and related options (like `selector_class`) will only be used if more than one connection is defined. Either directly or via the *Sniffing* mechanism.

5.3.1 Transport

```
class elasticsearch.Transport (hosts,                               connection_class=Urllib3HttpConnection,
                              connection_pool_class=ConnectionPool,
                              nodes_to_host_callback=construct_hosts_list, sniff_on_start=False,
                              sniffer_timeout=None,   sniff_on_connection_fail=False,   serial-
                              izer=JSONSerializer(), max_retries=3, **kwargs)
```

Encapsulation of transport-related to logic. Handles instantiation of the individual connections as well as creating a connection pool to hold them.

Main interface is the *perform_request* method.

Parameters

- **hosts** – list of dictionaries, each containing keyword arguments to create a *connection_class* instance
- **connection_class** – subclass of `Connection` to use
- **connection_pool_class** – subclass of `ConnectionPool` to use
- **host_info_callback** – callback responsible for taking the node information from */_cluster/nodes*, along with already extracted information, and producing a list of arguments (same as *hosts* parameter)
- **sniff_on_start** – flag indicating whether to obtain a list of nodes from the cluster at startup time
- **sniffer_timeout** – number of seconds between automatic sniffs
- **sniff_on_connection_fail** – flag controlling if connection failure triggers a sniff
- **sniff_timeout** – timeout used for the sniff request - it should be a fast api call and we are talking potentially to more nodes so we want to fail quickly. Not used during initial sniffing (if *sniff_on_start* is on) when the connection still isn't initialized.
- **serializer** – serializer instance
- **serializers** – optional dict of serializer instances that will be used for deserializing data coming from the server. (key is the mimetype)
- **default_mimetype** – when no mimetype is specified by the server response assume this mimetype, defaults to *'application/json'*
- **max_retries** – maximum number of retries before an exception is propagated
- **retry_on_status** – set of HTTP status codes on which we should retry on a different node. defaults to (503, 504,)
- **retry_on_timeout** – should timeout trigger a retry on different node? (default *False*)
- **send_get_body_as** – for GET requests with body this option allows you to specify an alternate way of execution for environments that don't support passing bodies with GET requests. If you set this to 'POST' a POST method will be used instead, if to 'source' then the body will be serialized and passed as a query parameter *source*.

Any extra keyword arguments will be passed to the *connection_class* when creating and instance unless overridden by that connection's options provided as part of the *hosts* parameter.

add_connection (*host*)

Create a new `Connection` instance and add it to the pool.

Parameters `host` – kwargs that will be used to create the instance

get_connection()

Retrieve a `Connection` instance from the `ConnectionPool` instance.

mark_dead(connection)

Mark a connection as dead (failed) in the connection pool. If sniffing on failure is enabled this will initiate the sniffing process.

Parameters `connection` – instance of `Connection` that failed

perform_request(method, url, params=None, body=None)

Perform the actual request. Retrieve a connection from the connection pool, pass all the information to it's `perform_request` method and return the data.

If an exception was raised, mark the connection as failed and retry (up to `max_retries` times).

If the operation was succesful and the connection used was previously marked as dead, mark it as live, resetting it's failure count.

Parameters

- **method** – HTTP method to use
- **url** – absolute url (without host) to target
- **params** – dictionary of query parameters, will be handed over to the underlying `Connection` class for serialization
- **body** – body of the request, will be serializes using serializer and passed to the connection

set_connections(hosts)

Instantiate all the connections and crate new connection pool to hold them. Tries to identify unchanged hosts and re-use existing `Connection` instances.

Parameters `hosts` – same as `__init__`

sniff_hosts(initial=False)

Obtain a list of nodes from the cluster and create a new connection pool using the information retrieved.

To extract the node connection parameters use the `nodes_to_host_callback`.

Parameters `initial` – flag indicating if this is during startup (`sniff_on_start`), ignore the `sniff_timeout` if `True`

5.3.2 Connection Pool

```
class elasticsearch.ConnectionPool(connections, dead_timeout=60, selector_class=RoundRobinSelector, randomize_hosts=True, **kwargs)
```

Container holding the `Connection` instances, managing the selection process (via a `ConnectionSelector`) and dead connections.

It's only interactions are with the `Transport` class that drives all the actions within `ConnectionPool`.

Initially connections are stored on the class as a list and, along with the connection options, get passed to the `ConnectionSelector` instance for future reference.

Upon each request the `Transport` will ask for a `Connection` via the `get_connection` method. If the connection fails (it's `perform_request` raises a `ConnectionError`) it will be marked as dead (via `mark_dead`) and put on a timeout (if it fails `N` times in a row the timeout is exponentially longer - the formula is `default_timeout * 2 ** (fail_count - 1)`). When the timeout is over the connection will be resurrected and returned to the live pool. A

connection that has been previously marked as dead and succeeds will be marked as live (it's fail count will be deleted).

Parameters

- **connections** – list of tuples containing the `Connection` instance and it's options
- **dead_timeout** – number of seconds a connection should be retired for after a failure, increases on consecutive failures
- **timeout_cutoff** – number of consecutive failures after which the timeout doesn't increase
- **selector_class** – `ConnectionSelector` subclass to use if more than one connection is live
- **randomize_hosts** – shuffle the list of connections upon arrival to avoid dog piling effect across processes

`get_connection()`

Return a connection from the pool using the `ConnectionSelector` instance.

It tries to resurrect eligible connections, forces a resurrection when no connections are available and passes the list of live connections to the selector instance to choose from.

Returns a connection instance and it's current fail count.

`mark_dead(connection, now=None)`

Mark the connection as dead (failed). Remove it from the live pool and put it on a timeout.

Parameters **connection** – the failed instance

`mark_live(connection)`

Mark connection as healthy after a resurrection. Resets the fail counter for the connection.

Parameters **connection** – the connection to redeem

`resurrect(force=False)`

Attempt to resurrect a connection from the dead pool. It will try to locate one (not all) eligible (it's timeout is over) connection to return to the live pool. Any resurrected connection is also returned.

Parameters **force** – resurrect a connection even if there is none eligible (used when we have no live connections). If force is specified resurrect always returns a connection.

5.3.3 Connection Selector

`class elasticsearch.ConnectionSelector(opts)`

Simple class used to select a connection from a list of currently live connection instances. In init time it is passed a dictionary containing all the connections' options which it can then use during the selection process. When the `select` method is called it is given a list of *currently* live connections to choose from.

The options dictionary is the one that has been passed to `Transport` as `hosts` param and the same that is used to construct the `Connection` object itself. When the `Connection` was created from information retrieved from the cluster via the sniffing process it will be the dictionary returned by the `host_info_callback`.

Example of where this would be useful is a zone-aware selector that would only select connections from it's own zones and only fall back to other connections where there would be none in it's zones.

Parameters **opts** – dictionary of connection instances and their options

`select(connections)`

Select a connection from the given list.

Parameters **connections** – list of live connections to choose from

5.3.4 Urllib3HttpConnection (default connection_class)

```
class elasticsearch.Urllib3HttpConnection (host='localhost', port=9200, http_auth=None,
                                           use_ssl=False, verify_certs=False, ca_certs=None,
                                           client_cert=None, ssl_version=None, maxsize=10,
                                           **kwargs)
```

Default connection class using the *urllib3* library and the http protocol.

Parameters

- **http_auth** – optional http auth information as either ':' separated string or a tuple
- **use_ssl** – use ssl for the connection if *True*
- **verify_certs** – whether to verify SSL certificates
- **ca_certs** – optional path to CA bundle. See <http://urllib3.readthedocs.org/en/latest/security.html#using-certifi-with-urllib3> for instructions how to get default set
- **client_cert** – path to the file containing the private key and the certificate
- **ssl_version** – version of the SSL protocol to use. Choices are: SSLv23 (default) SSLv2 SSLv3 TLSv1 (see `PROTOCOL_*` constants in the `ssl` module for exact options for your environment).
- **maxsize** – the maximum number of connections which will be kept open to this host.

5.4 Transport classes

List of transport classes that can be used, simply import your choice and pass it to the constructor of *Elasticsearch* as *connection_class*. Note that the *RequestsHttpConnection* requires *requests* to be installed.

For example to use the *requests*-based connection just import it and use it:

```
from elasticsearch import Elasticsearch, RequestsHttpConnection
es = Elasticsearch(connection_class=RequestsHttpConnection)
```

5.4.1 Connection

```
class elasticsearch.connection.Connection (host='localhost', port=9200, url_prefix='', timeout=10, **kwargs)
```

Class responsible for maintaining a connection to an Elasticsearch node. It holds persistent connection pool to it and it's main interface (*perform_request*) is thread-safe.

Also responsible for logging.

Parameters

- **host** – hostname of the node (default: localhost)
- **port** – port to use (integer, default: 9200)
- **url_prefix** – optional url prefix for elasticsearch
- **timeout** – default timeout in seconds (float, default: 10)

5.4.2 Urllib3HttpConnection

```
class elasticsearch.connection.Urllib3HttpConnection (host='localhost', port=9200,
                                                    http_auth=None, use_ssl=False,
                                                    verify_certs=False,
                                                    ca_certs=None, client_cert=None,
                                                    ssl_version=None, maxsize=10,
                                                    **kwargs)
```

Default connection class using the *urllib3* library and the http protocol.

Parameters

- **http_auth** – optional http auth information as either ':' separated string or a tuple
- **use_ssl** – use ssl for the connection if *True*
- **verify_certs** – whether to verify SSL certificates
- **ca_certs** – optional path to CA bundle. See <http://urllib3.readthedocs.org/en/latest/security.html#using-certifi-with-urllib3> for instructions how to get default set
- **client_cert** – path to the file containing the private key and the certificate
- **ssl_version** – version of the SSL protocol to use. Choices are: SSLv23 (default) SSLv2 SSLv3 TLSv1 (see `PROTOCOL_*` constants in the `ssl` module for exact options for your environment).
- **maxsize** – the maximum number of connections which will be kept open to this host.

5.4.3 RequestsHttpConnection

```
class elasticsearch.connection.RequestsHttpConnection (host='localhost',
                                                       port=9200, http_auth=None,
                                                       use_ssl=False, verify_certs=False,
                                                       ca_certs=None, client_cert=None,
                                                       **kwargs)
```

Connection using the *requests* library.

Parameters

- **http_auth** – optional http auth information as either ':' separated string or a tuple. Any value will be passed into requests as *auth*.
- **use_ssl** – use ssl for the connection if *True*
- **verify_certs** – whether to verify SSL certificates
- **ca_certs** – optional path to CA bundle. By default standard requests' bundle will be used.
- **client_cert** – path to the file containing the private key and the certificate

5.5 Helpers

Collection of simple helper functions that abstract some specifics or the raw API.

5.5.1 Bulk helpers

There are several helpers for the `bulk` API since it's requirement for specific formatting and other considerations can make it cumbersome if used directly.

All bulk helpers accept an instance of `Elasticsearch` class and an iterable `actions` (any iterable, can also be a generator, which is ideal in most cases since it will allow you to index large datasets without the need of loading them into memory).

The items in the `action` iterable should be the documents we wish to index in several formats. The most common one is the same as returned by `search()`, for example:

```
{
  '_index': 'index-name',
  '_type': 'document',
  '_id': 42,
  '_parent': 5,
  '_ttl': '1d',
  '_source': {
    "title": "Hello World!",
    "body": "... "
  }
}
```

Alternatively, if `_source` is not present, it will pop all metadata fields from the doc and use the rest as the document data:

```
{
  "_id": 42,
  "_parent": 5,
  "title": "Hello World!",
  "body": "... "
}
```

The `bulk()` api accepts index, create, delete, and update actions. Use the `_op_type` field to specify an action (`_op_type` defaults to index):

```
{
  '_op_type': 'delete',
  '_index': 'index-name',
  '_type': 'document',
  '_id': 42,
}
{
  '_op_type': 'update',
  '_index': 'index-name',
  '_type': 'document',
  '_id': 42,
  'doc': {'question': 'The life, universe and everything.'}
}
```

Note: When reading raw json strings from a file, you can also pass them in directly (without decoding to dicts first). In that case, however, you lose the ability to specify anything (index, type, even id) on a per-record basis, all documents will just be sent to elasticsearch to be indexed as-is.

```
elasticsearch.helpers.streaming_bulk(client, actions, chunk_size=500,
                                     max_chunk_bytes=103833600, raise_on_error=True,
                                     expand_action_callback=<function expand_action>,
                                     raise_on_exception=True, **kwargs)
```

Streaming bulk consumes actions from the iterable passed in and yields results per action. For non-streaming usecases use `bulk()` which is a wrapper around streaming bulk that returns summary information about the bulk operation once the entire input is consumed and sent.

Parameters

- **client** – instance of `Elasticsearch` to use
- **actions** – iterable containing the actions to be executed
- **chunk_size** – number of docs in one chunk sent to es (default: 500)
- **max_chunk_bytes** – the maximum size of the request in bytes (default: 100MB)
- **raise_on_error** – raise `BulkIndexError` containing errors (as `.errors`) from the execution of the last chunk when some occur. By default we raise.
- **raise_on_exception** – if `False` then don't propagate exceptions from call to `bulk` and just report the items that failed as failed.
- **expand_action_callback** – callback executed on each action passed in, should return a tuple containing the action line and the data line (`None` if data line should be omitted).

```
elasticsearch.helpers.parallel_bulk(client, actions, thread_count=4, chunk_size=500,
                                   max_chunk_bytes=103833600, expand_action_callback=<function
                                   expand_action>,
                                   **kwargs)
```

Parallel version of the bulk helper run in multiple threads at once.

Parameters

- **client** – instance of `Elasticsearch` to use
- **actions** – iterator containing the actions
- **thread_count** – size of the threadpool to use for the bulk requests
- **chunk_size** – number of docs in one chunk sent to es (default: 500)
- **max_chunk_bytes** – the maximum size of the request in bytes (default: 100MB)
- **raise_on_error** – raise `BulkIndexError` containing errors (as `.errors`) from the execution of the last chunk when some occur. By default we raise.
- **raise_on_exception** – if `False` then don't propagate exceptions from call to `bulk` and just report the items that failed as failed.
- **expand_action_callback** – callback executed on each action passed in, should return a tuple containing the action line and the data line (`None` if data line should be omitted).

```
elasticsearch.helpers.bulk(client, actions, stats_only=False, **kwargs)
```

Helper for the `bulk()` api that provides a more human friendly interface - it consumes an iterator of actions and sends them to elasticsearch in chunks. It returns a tuple with summary information - number of successfully executed actions and either list of errors or number of errors if `stats_only` is set to `True`.

See `streaming_bulk()` for more accepted parameters

Parameters

- **client** – instance of `Elasticsearch` to use
- **actions** – iterator containing the actions
- **stats_only** – if `True` only report number of successful/failed operations instead of just number of successful and a list of error responses

Any additional keyword arguments will be passed to `streaming_bulk()` which is used to execute the operation.

5.5.2 Scan

`elasticsearch.helpers.scan(client, query=None, scroll=u'5m', raise_on_error=True, preserve_order=False, **kwargs)`

Simple abstraction on top of the `scroll()` api - a simple iterator that yields all hits as returned by underlining scroll requests.

By default scan does not return results in any pre-determined order. To have a standard order in the returned documents (either by score or explicit sort definition) when scrolling, use `preserve_order=True`. This may be an expensive operation and will negate the performance benefits of using `scan`.

Parameters

- **client** – instance of `Elasticsearch` to use
- **query** – body for the `search()` api
- **scroll** – Specify how long a consistent view of the index should be maintained for scrolled search
- **raise_on_error** – raises an exception (`ScanError`) if an error is encountered (some shards fail to execute). By default we raise.
- **preserve_order** – don't set the `search_type` to `scan` - this will cause the scroll to paginate with preserving the order. Note that this can be an extremely expensive operation and can easily lead to unpredictable results, use with caution.

Any additional keyword arguments will be passed to the initial `search()` call:

```
scan(es,
    query={"match": {"title": "python"}},
    index="orders-*",
    doc_type="books"
)
```

5.5.3 Reindex

`elasticsearch.helpers.reindex(client, source_index, target_index, query=None, target_client=None, chunk_size=500, scroll=u'5m', scan_kwargs={}, bulk_kwargs={})`

Reindex all documents from one index that satisfy a given query to another, potentially (if `target_client` is specified) on a different cluster. If you don't specify the query you will reindex all the documents.

Note: This helper doesn't transfer mappings, just the data.

Parameters

- **client** – instance of `Elasticsearch` to use (for read if `target_client` is specified as well)
- **source_index** – index (or list of indices) to read documents from
- **target_index** – name of the index in the target cluster to populate
- **query** – body for the `search()` api

- **target_client** – optional, is specified will be used for writing (thus enabling reindex between clusters)
- **chunk_size** – number of docs in one chunk sent to es (default: 500)
- **scroll** – Specify how long a consistent view of the index should be maintained for scrolled search
- **scan_kwargs** – additional kwargs to be passed to `scan()`
- **bulk_kwargs** – additional kwargs to be passed to `bulk()`

5.6 Changelog

5.6.1 2.0.0 (2015-10-14)

- Elasticsearch 2.0 compatibility release

5.6.2 1.8.0 (2015-10-14)

- removed thrift and memcached connections, if you wish to continue using those, extract the classes and use them separately.
- added a new, parallel version of the bulk helper using thread pools

5.6.3 1.7.0 (2015-09-21)

- elasticsearch 2.0 compatibility
- thrift now deprecated, to be removed in future version
- make sure urllib3 always uses keep-alive

5.6.4 1.6.0 (2015-06-10)

- Add `indices.flush_synced` API
- `helpers.reindex` now supports reindexing parent/child documents

5.6.5 1.5.0 (2015-05-18)

- Add support for `query_cache` parameter when searching
- helpers have been made more secure by changing defaults to raise an exception on errors
- removed deprecated options `replication` and the deprecated benchmark api.
- Added `AddonClient` class to allow for extending the client from outside

5.6.6 1.4.0 (2015-02-11)

- Using insecure SSL configuration (`verify_cert=False`) raises a warning
- `reindex` accepts a `query` parameter
- enable `reindex` helper to accept any kwargs for underlying `bulk` and `scan` calls
- when doing an initial sniff (via `sniff_on_start`) ignore special sniff timeout
- option to treat `TransportError` as normal failure in `bulk` helpers
- fixed an issue with sniffing when only a single host was passed in

5.6.7 1.3.0 (2014-12-31)

- Timeout now doesn't trigger a retry by default (can be overridden by setting `retry_on_timeout=True`)
- Introduced new parameter `retry_on_status` (defaulting to `(503, 504,)`) controls which http status code should lead to a retry.
- Implemented url parsing according to RFC-1738
- Added support for proper SSL certificate handling
- Required parameters are now checked for non-empty values
- `ConnectionPool` now checks if any connections were defined
- `DummyConnectionPool` introduced when no load balancing is needed (only one connection defined)
- Fixed a race condition in `ConnectionPool`

5.6.8 1.2.0 (2014-08-03)

Compatibility with newest (1.3) Elasticsearch APIs.

- Filter out master-only nodes when sniffing
- Improved docs and error messages

5.6.9 1.1.1 (2014-07-04)

Bugfix release fixing escaping issues with `request_timeout`.

5.6.10 1.1.0 (2014-07-02)

Compatibility with newest Elasticsearch APIs.

- Test helpers - `ElasticsearchTestCase` and `get_test_client` for use in your tests
- Python 3.2 compatibility
- Use `simplejson` if installed instead of `stdlib json` library
- Introducing a global `request_timeout` parameter for per-call timeout
- Bug fixes

5.6.11 1.0.0 (2014-02-11)

Elasticsearch 1.0 compatibility. See 0.4.X releases (and 0.4 branch) for code compatible with 0.90 elasticsearch.

- major breaking change - compatible with 1.0 elasticsearch releases only!
- Add an option to change the timeout used for sniff requests (`sniff_timeout`).
- empty responses from the server are now returned as empty strings instead of `None`
- `get_alias` now has `name` as another optional parameter due to issue #4539 in es repo. Note that the order of params have changed so if you are not using keyword arguments this is a breaking change.

5.6.12 0.4.4 (2013-12-23)

- `helpers.bulk_index` renamed to `helpers.bulk` (alias put in place for backwards compatibility, to be removed in future versions)
- Added `helpers.streaming_bulk` to consume an iterator and yield results per operation
- `helpers.bulk` and `helpers.streaming_bulk` are no longer limited to just index operations.
- unicode body (for `indices.analyze` for example) is now handled correctly
- changed `perform_request` on `Connection` classes to return headers as well. This is a backwards incompatible change for people who have developed their own connection class.
- changed deserialization mechanics. Users who provided their own serializer that didn't extend `JSONSerializer` need to specify a `mimetype` class attribute.
- minor bug fixes

5.6.13 0.4.3 (2013-10-22)

- Fixes to `helpers.bulk_index`, better error handling
- More benevolent `hosts` argument parsing for `Elasticsearch`
- `requests` no longer required (nor recommended) for install

5.6.14 0.4.2 (2013-10-08)

- `ignore` param acceted by all APIs
- Fixes to `helpers.bulk_index`

5.6.15 0.4.1 (2013-09-24)

Initial release.

License

Copyright 2013 Elasticsearch

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Indices and tables

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

e

`elasticsearch`, [53](#)
`elasticsearch.client`, [30](#)
`elasticsearch.connection`, [57](#)
`elasticsearch.helpers`, [59](#)

A

`add_connection()` (elasticsearch.Transport method), 54
`aliases()` (elasticsearch.client.CatClient method), 46
`allocation()` (elasticsearch.client.CatClient method), 47
`analyze()` (elasticsearch.client.IndicesClient method), 30

B

`bulk()` (elasticsearch.Elasticsearch method), 15
`bulk()` (in module elasticsearch.helpers), 60

C

`CatClient` (class in elasticsearch.client), 46
`clear_cache()` (elasticsearch.client.IndicesClient method), 31
`clear_scroll()` (elasticsearch.Elasticsearch method), 15
`close()` (elasticsearch.client.IndicesClient method), 31
`ClusterClient` (class in elasticsearch.client), 43
`ConflictError` (class in elasticsearch), 53
`Connection` (class in elasticsearch.connection), 57
`ConnectionError` (class in elasticsearch), 53
`ConnectionPool` (class in elasticsearch), 55
`ConnectionSelector` (class in elasticsearch), 56
`ConnectionTimeout` (class in elasticsearch), 53
`count()` (elasticsearch.client.CatClient method), 47
`count()` (elasticsearch.Elasticsearch method), 15
`count_percolate()` (elasticsearch.Elasticsearch method), 16
`create()` (elasticsearch.client.IndicesClient method), 31
`create()` (elasticsearch.client.SnapshotClient method), 51
`create()` (elasticsearch.Elasticsearch method), 17
`create_repository()` (elasticsearch.client.SnapshotClient method), 51

D

`delete()` (elasticsearch.client.IndicesClient method), 32
`delete()` (elasticsearch.client.SnapshotClient method), 51
`delete()` (elasticsearch.Elasticsearch method), 17
`delete_alias()` (elasticsearch.client.IndicesClient method), 32

`delete_repository()` (elasticsearch.client.SnapshotClient method), 51
`delete_script()` (elasticsearch.Elasticsearch method), 18
`delete_template()` (elasticsearch.client.IndicesClient method), 32
`delete_template()` (elasticsearch.Elasticsearch method), 18
`delete_warmer()` (elasticsearch.client.IndicesClient method), 32

E

`Elasticsearch` (class in elasticsearch), 14
`elasticsearch` (module), 14, 53, 54
`elasticsearch.client` (module), 30
`elasticsearch.connection` (module), 57
`elasticsearch.helpers` (module), 59
`ElasticsearchException` (class in elasticsearch), 53
`error` (elasticsearch.TransportError attribute), 53
`exists()` (elasticsearch.client.IndicesClient method), 32
`exists()` (elasticsearch.Elasticsearch method), 18
`exists_alias()` (elasticsearch.client.IndicesClient method), 33
`exists_template()` (elasticsearch.client.IndicesClient method), 33
`exists_type()` (elasticsearch.client.IndicesClient method), 33
`explain()` (elasticsearch.Elasticsearch method), 19

F

`field_stats()` (elasticsearch.Elasticsearch method), 19
`fielddata()` (elasticsearch.client.CatClient method), 47
`flush()` (elasticsearch.client.IndicesClient method), 34
`flush_synced()` (elasticsearch.client.IndicesClient method), 34

G

`get()` (elasticsearch.client.IndicesClient method), 34
`get()` (elasticsearch.client.SnapshotClient method), 51
`get()` (elasticsearch.Elasticsearch method), 20
`get_alias()` (elasticsearch.client.IndicesClient method), 35

get_aliases() (elasticsearch.client.IndicesClient method), 35

get_connection() (elasticsearch.ConnectionPool method), 56

get_connection() (elasticsearch.Transport method), 55

get_field_mapping() (elasticsearch.client.IndicesClient method), 35

get_mapping() (elasticsearch.client.IndicesClient method), 36

get_repository() (elasticsearch.client.SnapshotClient method), 52

get_script() (elasticsearch.Elasticsearch method), 20

get_settings() (elasticsearch.client.ClusterClient method), 43

get_settings() (elasticsearch.client.IndicesClient method), 36

get_source() (elasticsearch.Elasticsearch method), 20

get_template() (elasticsearch.client.IndicesClient method), 36

get_template() (elasticsearch.Elasticsearch method), 21

get_upgrade() (elasticsearch.client.IndicesClient method), 37

get_warmer() (elasticsearch.client.IndicesClient method), 37

H

health() (elasticsearch.client.CatClient method), 48

health() (elasticsearch.client.ClusterClient method), 43

help() (elasticsearch.client.CatClient method), 48

hot_threads() (elasticsearch.client.NodesClient method), 45

I

ImproperlyConfigured (class in elasticsearch), 53

index() (elasticsearch.Elasticsearch method), 21

indices() (elasticsearch.client.CatClient method), 48

IndicesClient (class in elasticsearch.client), 30

info (elasticsearch.TransportError attribute), 53

info() (elasticsearch.client.NodesClient method), 46

info() (elasticsearch.Elasticsearch method), 22

M

mark_dead() (elasticsearch.ConnectionPool method), 56

mark_dead() (elasticsearch.Transport method), 55

mark_live() (elasticsearch.ConnectionPool method), 56

master() (elasticsearch.client.CatClient method), 48

mget() (elasticsearch.Elasticsearch method), 22

mpercolate() (elasticsearch.Elasticsearch method), 22

msearch() (elasticsearch.Elasticsearch method), 22

termvectors() (elasticsearch.Elasticsearch method), 23

N

nodeattrs() (elasticsearch.client.CatClient method), 49

nodes() (elasticsearch.client.CatClient method), 49

NodesClient (class in elasticsearch.client), 45

NotFoundError (class in elasticsearch), 53

O

open() (elasticsearch.client.IndicesClient method), 37

optimize() (elasticsearch.client.IndicesClient method), 38

P

parallel_bulk() (in module elasticsearch.helpers), 60

pending_tasks() (elasticsearch.client.CatClient method), 49

pending_tasks() (elasticsearch.client.ClusterClient method), 44

percolate() (elasticsearch.Elasticsearch method), 23

perform_request() (elasticsearch.Transport method), 55

ping() (elasticsearch.Elasticsearch method), 24

plugins() (elasticsearch.client.CatClient method), 49

put_alias() (elasticsearch.client.IndicesClient method), 38

put_mapping() (elasticsearch.client.IndicesClient method), 38

put_script() (elasticsearch.Elasticsearch method), 24

put_settings() (elasticsearch.client.ClusterClient method), 44

put_settings() (elasticsearch.client.IndicesClient method), 39

put_template() (elasticsearch.client.IndicesClient method), 39

put_template() (elasticsearch.Elasticsearch method), 25

put_warmer() (elasticsearch.client.IndicesClient method), 40

R

recovery() (elasticsearch.client.CatClient method), 50

recovery() (elasticsearch.client.IndicesClient method), 40

refresh() (elasticsearch.client.IndicesClient method), 40

reindex() (in module elasticsearch.helpers), 61

render_search_template() (elasticsearch.Elasticsearch method), 25

RequestError (class in elasticsearch), 53

RequestsHttpConnection (class in elasticsearch.connection), 58

reroute() (elasticsearch.client.ClusterClient method), 44

restore() (elasticsearch.client.SnapshotClient method), 52

resurrect() (elasticsearch.ConnectionPool method), 56

S

scan() (in module elasticsearch.helpers), 61

scroll() (elasticsearch.Elasticsearch method), 25

search() (elasticsearch.Elasticsearch method), 25

search_exists() (elasticsearch.Elasticsearch method), 27

search_shards() (elasticsearch.Elasticsearch method), 27

search_template() (elasticsearch.Elasticsearch method), 28

segments() (elasticsearch.client.CatClient method), 50

[segments\(\)](#) (elasticsearch.client.IndicesClient method), [41](#)
[select\(\)](#) (elasticsearch.ConnectionSelector method), [56](#)
[SerializationError](#) (class in elasticsearch), [53](#)
[set_connections\(\)](#) (elasticsearch.Transport method), [55](#)
[shard_stores\(\)](#) (elasticsearch.client.IndicesClient method), [41](#)
[shards\(\)](#) (elasticsearch.client.CatClient method), [50](#)
[SnapshotClient](#) (class in elasticsearch.client), [51](#)
[sniff_hosts\(\)](#) (elasticsearch.Transport method), [55](#)
[SSLError](#) (class in elasticsearch), [53](#)
[state\(\)](#) (elasticsearch.client.ClusterClient method), [44](#)
[stats\(\)](#) (elasticsearch.client.ClusterClient method), [45](#)
[stats\(\)](#) (elasticsearch.client.IndicesClient method), [41](#)
[stats\(\)](#) (elasticsearch.client.NodesClient method), [46](#)
[status\(\)](#) (elasticsearch.client.SnapshotClient method), [52](#)
[status_code](#) (elasticsearch.TransportError attribute), [53](#)
[streaming_bulk\(\)](#) (in module elasticsearch.helpers), [59](#)
[suggest\(\)](#) (elasticsearch.Elasticsearch method), [28](#)

T

[termvectors\(\)](#) (elasticsearch.Elasticsearch method), [29](#)
[thread_pool\(\)](#) (elasticsearch.client.CatClient method), [50](#)
[Transport](#) (class in elasticsearch), [54](#)
[TransportError](#) (class in elasticsearch), [53](#)

U

[update\(\)](#) (elasticsearch.Elasticsearch method), [29](#)
[update_aliases\(\)](#) (elasticsearch.client.IndicesClient method), [42](#)
[upgrade\(\)](#) (elasticsearch.client.IndicesClient method), [42](#)
[Urllib3HttpConnection](#) (class in elasticsearch), [57](#)
[Urllib3HttpConnection](#) (class in elasticsearch.connection), [58](#)

V

[validate_query\(\)](#) (elasticsearch.client.IndicesClient method), [42](#)
[verify_repository\(\)](#) (elasticsearch.client.SnapshotClient method), [52](#)