Elasticsearch Documentation

Release 7.5.1

Honza Král

Jan 23, 2020

Contents

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`.

CHAPTER 1

Compatibility

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

For **Elasticsearch 7.0** and later, use the major version 7 (`7.x.y`) of the library.

For **Elasticsearch 6.0** and later, use the major version 6 (`6.x.y`) of the library.

For **Elasticsearch 5.0** and later, use the major version 5 (`5.x.y`) of the library.

For **Elasticsearch 2.0** and later, use the major version 2 (`2.x.y`) of the library, and so on.

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

```
# Elasticsearch 7.x
elasticsearch>=7.0.0,<8.0.0

# Elasticsearch 6.x
elasticsearch>=6.0.0,<7.0.0

# Elasticsearch 5.x
elasticsearch>=5.0.0,<6.0.0

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

If you have a need to have multiple versions installed at the same time older versions are also released as `elasticsearch2`, `elasticsearch5` and `elasticsearch6`.

CHAPTER 2

Installation

Install the `elasticsearch` package with pip:

```
pip install elasticsearch
```

CHAPTER 3

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", id=1, body=doc)
print(res['result'])

res = es.get(index="test-index", 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']['value'])
for hit in res['hits']['hits']:
    print("%(timestamp)s %(author)s: %(text)s" % hit["_source"])
```

CHAPTER 4

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 as well as a more high level library (elasticsearch-dsl) on top of this one to provide a more convenient way of working with Elasticsearch.

4.1 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 between two `http` protocol implementations. 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.

4.2 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.

4.3 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_timeout=60)
```

4.4 Thread safety

The client is thread safe and can be used in a multi threaded environment. Best practice is to create a single global instance of the client and use it throughout your application. If your application is long-running consider turning on *Sniffing* to make sure the client is up to date on the cluster location.

By default we allow `urllib3` to open up to 10 connections to each node, if your application calls for more parallelism, use the `maxsize` parameter to raise the limit:

```
# allow up to 25 connections to each node
es = Elasticsearch(["host1", "host2"], maxsize=25)
```

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`. Note that Python's `multiprocessing` module uses `fork` to create new processes on POSIX systems.

4.5 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
```

(continues on next page)

(continued from previous page)

```
es = Elasticsearch(['https://user:secret@localhost:443'])

# ... or specify common parameters as kwargs

es = Elasticsearch(
    ['localhost', 'otherhost'],
    http_auth=('user', 'secret'),
    scheme="https",
    port=443,
)

# SSL client authentication using client_cert and client_key

from ssl import create_default_context

context = create_default_context(cafile="path/to/cert.pem")
es = Elasticsearch(
    ['localhost', 'otherhost'],
    http_auth=('user', 'secret'),
    scheme="https",
    port=443,
    ssl_context=context,
)
```

Warning: `elasticsearch-py` doesn't ship with default set of root certificates. To have working SSL certificate validation you need to either specify your own as `cafile` or `capath` or `cadata` or install certifi which will be picked up automatically.

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

4.6 APIKey Authentication

You can configure the client to use Elasticsearch's API Key for connecting to your cluster. Please note this authentication method has been introduced with release of Elasticsearch `6.7.0`.

from elasticsearch import Elasticsearch

you can use the api key tuple es = Elasticsearch(

['node-1', 'node-2', 'node-3'], api_key=('id', 'api_key'),

)

or you pass the base 64 encoded token es = Elasticsearch(

['node-1', 'node-2', 'node-3'], api_key='base64encoded tuple',

)

4.7 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. Because it is designed to be shared (for example to demonstrate an issue) it also just uses `localhost:9200` as the address instead of the actual address of the host. If the trace logger has not been configured already it is set to *propagate=False* so it needs to be activated separately.

CHAPTER 5

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')
```

5.1 Compression

When using capacity-constrained networks (low throughput), it may be handy to enable compression. This is especially useful when doing bulk loads or inserting large documents. This will configure compression.

```
from elasticsearch import Elasticsearch
es = Elasticsearch(hosts, http_compress=True)
```

5.2 Running on AWS with IAM

If you want to use this client with IAM based authentication on AWS you can use the requests-aws4auth package:

```
from elasticsearch import Elasticsearch, RequestsHttpConnection
from requests_aws4auth import AWS4Auth

host = 'YOURHOST.us-east-1.es.amazonaws.com'
awsauth = AWS4Auth(YOUR_ACCESS_KEY, YOUR_SECRET_KEY, REGION, 'es')
```

(continues on next page)

(continued from previous page)

```
es = Elasticsearch(
    hosts=[{'host': host, 'port': 443}],
    http_auth=awsauth,
    use_ssl=True,
    verify_certs=True,
    connection_class=RequestsHttpConnection
)
print(es.info())
```

CHAPTER 6

Customization

6.1 Custom serializers

By default, JSONSerializer is used to encode all outgoing requests. However, you can implement your own custom serializer:

```
from elasticsearch.serializer import JSONSerializer

class SetEncoder(JSONSerializer):
    def default(self, obj):
        if isinstance(obj, set):
            return list(obj)
        if isinstance(obj, Something):
            return 'CustomSomethingRepresentation'
        return JSONSerializer.default(self, obj)

es = Elasticsearch(serializer=SetEncoder())
```

CHAPTER 7

Contents

7.1 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.

7.1.1 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.

Response Filtering

The `filter_path` parameter is used to reduce the response returned by elasticsearch. For example, to only return `_id` and `_type`, do:

```
es.search(index='test-index', filter_path=['hits.hits._id', 'hits.hits._type'])
```

It also supports the `*` wildcard character to match any field or part of a field's name:

```
es.search(index='test-index', filter_path=['hits.hits._*'])
```

7.1.2 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`, `ingest`, `nodes`, `snapshot` and `tasks` that provide access to instances of *CatClient*, *ClusterClient*, *IndicesClient*, *IngestClient*, *NodesClient*, *SnapshotClient* and *TasksClient* 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
```

(continues on next page)

(continued from previous page)

```
    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
    verify_certs=True,
    # provide a path to CA certs on disk
    ca_certs='/path/to/CA_certs'
)
```

If using SSL, but don't verify the certs, a warning message is showed optionally (see *Urllib3HttpConnection* for detailed description of the options):

```
es = Elasticsearch(
    ['localhost:443', 'other_host:443'],
    # turn on SSL
    use_ssl=True,
    # no verify SSL certificates
    verify_certs=False,
    # don't show warnings about ssl certs verification
    ssl_show_warn=False
)
```

SSL client authentication is supported (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
    verify_certs=True,
    # provide a path to CA certs on disk
    ca_certs='/path/to/CA_certs',
    # PEM formatted SSL client certificate
    client_cert='/path/to/clientcert.pem',
    # PEM formatted SSL client key
    client_key='/path/to/clientkey.pem'
)
```

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
)
```

By default, JSONSerializer is used to encode all outgoing requests. However, you can implement your own custom serializer:

```
from elasticsearch.serializer import JSONSerializer

class SetEncoder(JSONSerializer):
    def default(self, obj):
        if isinstance(obj, set):
            return list(obj)
        if isinstance(obj, Something):
            return 'CustomSomethingRepresentation'
        return JSONSerializer.default(self, obj)

es = Elasticsearch(serializer=SetEncoder())
```

Parameters

- **hosts** – list of nodes, or a single node, 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 `Connection` 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(***kwargs*)

Allows to perform multiple index/update/delete operations in a single request. https://www.elastic.co/guide/en/elasticsearch/reference/master/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
- **_source** – True or false to return the _source field or not, or default list of fields to return, can be overridden on each sub- request
- **_source_excludes** – Default list of fields to exclude from the returned _source field, can be overridden on each sub-request
- **_source_includes** – Default list of fields to extract and return from the _source field, can be overridden on each sub-request
- **doc_type** – Default document type for items which don’t provide one
- **pipeline** – The pipeline id to preprocess incoming documents with

- **refresh** – If *true* then refresh the effected shards to make this operation visible to search, if *wait_for* then wait for a refresh to make this operation visible to search, if *false* (the default) then do nothing with refreshes. Valid choices: true, false, wait_for
- **routing** – Specific routing value
- **timeout** – Explicit operation timeout
- **wait_for_active_shards** – Sets the number of shard copies that must be active before proceeding with the bulk operation. Defaults to 1, meaning the primary shard only. Set to *all* for all shard copies, otherwise set to any non-negative value less than or equal to the total number of copies for the shard (number of replicas + 1)

clear_scroll(***kwargs*)

Explicitly clears the search context for a scroll. https://www.elastic.co/guide/en/elasticsearch/reference/master/search-request-body.html#_clear_scroll_api

Parameters

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

count(***kwargs*)

Returns number of documents matching a query. https://www.elastic.co/guide/en/elasticsearch/reference/master/search-count.html

Parameters

- **body** – A query to restrict the results specified with the Query DSL (optional)
- **index** – A comma-separated list of indices to restrict the results
- **doc_type** – A comma-separated list of types to restrict the results
- **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) Valid choices: AND, OR Default: 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. Valid choices: open, closed, none, all Default: open
- **ignore_throttled** – Whether specified concrete, expanded or aliased indices should be ignored when throttled
- **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
- **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** – A comma-separated list of specific routing values
- **terminate_after** – The maximum count for each shard, upon reaching which the query execution will terminate early

create(***kwargs*)

Creates a new document in the index. Returns a 409 response when a document with a same ID already exists in the index. https://www.elastic.co/guide/en/elasticsearch/reference/master/docs-index_.html

Parameters

- **index** – The name of the index
- **id** – Document ID
- **body** – The document
- **doc_type** – The type of the document
- **pipeline** – The pipeline id to preprocess incoming documents with
- **refresh** – If *true* then refresh the affected shards to make this operation visible to search, if *wait_for* then wait for a refresh to make this operation visible to search, if *false* (the default) then do nothing with refreshes. Valid choices: true, false, wait_for
- **routing** – Specific routing value
- **timeout** – Explicit operation timeout
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type Valid choices: internal, external, external_gte
- **wait_for_active_shards** – Sets the number of shard copies that must be active before proceeding with the index operation. Defaults to 1, meaning the primary shard only. Set to *all* for all shard copies, otherwise set to any non-negative value less than or equal to the total number of copies for the shard (number of replicas + 1)

delete(***kwargs*)

Removes a document from the index. https://www.elastic.co/guide/en/elasticsearch/reference/master/docs-delete.html

Parameters

- **index** – The name of the index
- **id** – The document ID
- **doc_type** – The type of the document
- **if_primary_term** – only perform the delete operation if the last operation that has changed the document has the specified primary term
- **if_seq_no** – only perform the delete operation if the last operation that has changed the document has the specified sequence number
- **refresh** – If *true* then refresh the effected shards to make this operation visible to search, if *wait_for* then wait for a refresh to make this operation visible to search, if *false* (the default) then do nothing with refreshes. Valid choices: true, false, wait_for
- **routing** – Specific routing value
- **timeout** – Explicit operation timeout
- **version** – Explicit version number for concurrency control

- **version_type** – Specific version type Valid choices: internal, external, external_gte, force
- **wait_for_active_shards** – Sets the number of shard copies that must be active before proceeding with the delete operation. Defaults to 1, meaning the primary shard only. Set to *all* for all shard copies, otherwise set to any non-negative value less than or equal to the total number of copies for the shard (number of replicas + 1)

delete_by_query(***kwargs*)

Deletes documents matching the provided query. https://www.elastic.co/guide/en/elasticsearch/reference/master/docs-delete-by-query.html

Parameters

- **index** – A comma-separated list of index names to search; use *_all* or empty string to perform the operation on all indices
- **body** – The search definition using the Query DSL
- **doc_type** – A comma-separated list of document types to search; leave empty to perform the operation on all types
- **_source** – True or false to return the _source field or not, or a list of fields to return
- **_source_excludes** – A list of fields to exclude from the returned _source field
- **_source_includes** – 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)
- **conflicts** – What to do when the delete by query hits version conflicts? Valid choices: abort, proceed Default: abort
- **default_operator** – The default operator for query string query (AND or OR) Valid choices: AND, OR Default: 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. Valid choices: open, closed, none, all Default: open
- **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
- **max_docs** – Maximum number of documents to process (default: all documents)
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **q** – Query in the Lucene query string syntax
- **refresh** – Should the effected indexes be refreshed?
- **request_cache** – Specify if request cache should be used for this request or not, defaults to index level setting

- **requests_per_second** – The throttle for this request in sub- requests per second. -1 means no throttle.
- **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
- **scroll_size** – Size on the scroll request powering the delete by query
- **search_timeout** – Explicit timeout for each search request. Defaults to no timeout.
- **search_type** – Search operation type Valid choices: query_then_fetch, dfs_query_then_fetch
- **size** – Deprecated, please use *max_docs* instead
- **slices** – The number of slices this task should be divided into. Defaults to 1 meaning the task isn't sliced into subtasks. Default: 1
- **sort** – A comma-separated list of <field>:<direction> pairs
- **stats** – Specific 'tag' of the request for logging and statistical purposes
- **terminate_after** – The maximum number of documents to collect for each shard, upon reaching which the query execution will terminate early.
- **timeout** – Time each individual bulk request should wait for shards that are unavailable. Default: 1m
- **version** – Specify whether to return document version as part of a hit
- **wait_for_active_shards** – Sets the number of shard copies that must be active before proceeding with the delete by query operation. Defaults to 1, meaning the primary shard only. Set to *all* for all shard copies, otherwise set to any non-negative value less than or equal to the total number of copies for the shard (number of replicas + 1)
- **wait_for_completion** – Should the request should block until the delete by query is complete. Default: True

delete_by_query_rethrottle(***kwargs*)
Changes the number of requests per second for a particular Delete By Query operation. https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-delete-by-query.html

Parameters

- **task_id** – The task id to rethrottle
- **requests_per_second** – The throttle to set on this request in floating sub-requests per second. -1 means set no throttle.

delete_script(***kwargs*)
Deletes a script. https://www.elastic.co/guide/en/elasticsearch/reference/master/modules-scripting.html

Parameters

- **id** – Script ID
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit operation timeout

exists(***kwargs*)
Returns information about whether a document exists in an index. https://www.elastic.co/guide/en/elasticsearch/reference/master/docs-get.html

Parameters

- **index** – The name of the index
- **id** – The document ID
- **doc_type** – The type of the document (use *_all* to fetch the first document matching the ID across all types)
- **_source** – True or false to return the _source field or not, or a list of fields to return
- **_source_excludes** – A list of fields to exclude from the returned _source field
- **_source_includes** – A list of fields to extract and return from the _source field
- **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
- **stored_fields** – A comma-separated list of stored fields to return in the response
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type Valid choices: internal, external, external_gte, force

exists_source(***kwargs*)

Returns information about whether a document source exists in an index. https://www.elastic.co/guide/en/elasticsearch/reference/master/docs-get.html

Parameters

- **index** – The name of the index
- **id** – The document ID
- **doc_type** – The type of the document; deprecated and optional starting with 7.0
- **_source** – True or false to return the _source field or not, or a list of fields to return
- **_source_excludes** – A list of fields to exclude from the returned _source field
- **_source_includes** – A list of fields to extract and return from the _source field
- **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: internal, external, external_gte, force

explain(***kwargs*)

Returns information about why a specific matches (or doesn't match) a query. https://www.elastic.co/guide/en/elasticsearch/reference/master/search-explain.html

Parameters

- **index** – The name of the index
- **id** – The document ID
- **body** – The query definition using the Query DSL
- **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_excludes** – A list of fields to exclude from the returned _source field
- **_source_includes** – 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) Valid choices: AND, OR Default: OR
- **df** – The default field for query string query (default: _all)
- **lenient** – Specify whether format-based query failures (such as providing text to a numeric field) should be ignored
- **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
- **stored_fields** – A comma-separated list of stored fields to return in the response

field_caps(***kwargs*)

Returns the information about the capabilities of fields among multiple indices. https://www.elastic.co/guide/en/elasticsearch/reference/master/search-field-caps.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. Valid choices: open, closed, none, all Default: open
- **fields** – A comma-separated list of field names
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **include_unmapped** – Indicates whether unmapped fields should be included in the response.

get(***kwargs*)

Returns a document. https://www.elastic.co/guide/en/elasticsearch/reference/master/docs-get.html

Parameters

- **index** – The name of the index
- **id** – The document ID

- **doc_type** – The type of the document (use *_all* to fetch the first document matching the ID across all types)
- **_source** – True or false to return the _source field or not, or a list of fields to return
- **_source_excludes** – A list of fields to exclude from the returned _source field
- **_source_includes** – A list of fields to extract and return from the _source field
- **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
- **stored_fields** – A comma-separated list of stored fields to return in the response
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type Valid choices: internal, external, external_gte, force

get_script (***kwargs*)
Returns a script. https://www.elastic.co/guide/en/elasticsearch/reference/master/modules-scripting.html

Parameters

- **id** – Script ID
- **master_timeout** – Specify timeout for connection to master

get_source (***kwargs*)
Returns the source of a document. https://www.elastic.co/guide/en/elasticsearch/reference/master/docs-get.html

Parameters

- **index** – The name of the index
- **id** – The document ID
- **doc_type** – The type of the document; deprecated and optional starting with 7.0
- **_source** – True or false to return the _source field or not, or a list of fields to return
- **_source_excludes** – A list of fields to exclude from the returned _source field
- **_source_includes** – A list of fields to extract and return from the _source field
- **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: internal, external, external_gte, force

index (***kwargs*)
Creates or updates a document in an index. https://www.elastic.co/guide/en/elasticsearch/reference/master/docs-index_.html

Parameters

- **index** – The name of the index
- **body** – The document
- **doc_type** – The type of the document
- **id** – Document ID
- **if_primary_term** – only perform the index operation if the last operation that has changed the document has the specified primary term
- **if_seq_no** – only perform the index operation if the last operation that has changed the document has the specified sequence number
- **op_type** – Explicit operation type. Defaults to *index* for requests with an explicit document ID, and to 'create'for requests without an explicit document ID Valid choices: index, create
- **pipeline** – The pipeline id to preprocess incoming documents with
- **refresh** – If *true* then refresh the affected shards to make this operation visible to search, if *wait_for* then wait for a refresh to make this operation visible to search, if *false* (the default) then do nothing with refreshes. Valid choices: true, false, wait_for
- **routing** – Specific routing value
- **timeout** – Explicit operation timeout
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type Valid choices: internal, external, external_gte
- **wait_for_active_shards** – Sets the number of shard copies that must be active before proceeding with the index operation. Defaults to 1, meaning the primary shard only. Set to *all* for all shard copies, otherwise set to any non-negative value less than or equal to the total number of copies for the shard (number of replicas + 1)

info (***kwargs*)
Returns basic information about the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html

mget (***kwargs*)
Allows to get multiple documents in one request. https://www.elastic.co/guide/en/elasticsearch/reference/master/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_excludes** – A list of fields to exclude from the returned _source field
- **_source_includes** – A list of fields to extract and return from the _source field

- **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
- **stored_fields** – A comma-separated list of stored fields to return in the response

msearch(***kwargs*)

Allows to execute several search operations in one request. https://www.elastic.co/guide/en/elasticsearch/reference/master/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
- **ccs_minimize_roundtrips** – Indicates whether network round- trips should be minimized as part of cross-cluster search requests execution Default: true
- **max_concurrent_searches** – Controls the maximum number of concurrent searches the multi search api will execute
- **max_concurrent_shard_requests** – The number of concurrent shard requests each sub search executes concurrently per node. This value should be used to limit the impact of the search on the cluster in order to limit the number of concurrent shard requests Default: 5
- **pre_filter_shard_size** – A threshold that enforces a pre- filter roundtrip to pre-filter search shards based on query rewriting if the number of shards the search request expands to exceeds the threshold. This filter roundtrip can limit the number of shards significantly if for instance a shard can not match any documents based on it's rewrite method ie. if date filters are mandatory to match but the shard bounds and the query are disjoint. Default: 128
- **rest_total_hits_as_int** – Indicates whether hits.total should be rendered as an integer or an object in the rest search response
- **search_type** – Search operation type Valid choices: query_then_fetch, query_and_fetch, dfs_query_then_fetch, dfs_query_and_fetch
- **typed_keys** – Specify whether aggregation and suggester names should be prefixed by their respective types in the response

msearch_template(***kwargs*)

Allows to execute several search template operations in one request. https://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

- **max_concurrent_searches** – Controls the maximum number of concurrent searches the multi search api will execute
- **rest_total_hits_as_int** – Indicates whether hits.total should be rendered as an integer or an object in the rest search response
- **search_type** – Search operation type Valid choices: query_then_fetch, query_and_fetch, dfs_query_then_fetch, dfs_query_and_fetch
- **typed_keys** – Specify whether aggregation and suggester names should be prefixed by their respective types in the response

mtermvectors(***kwargs*)

Returns multiple termvectors in one request. https://www.elastic.co/guide/en/elasticsearch/reference/master/docs-multi-termvectors.html

Parameters

- **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.
- **index** – The index in which the document resides.
- **doc_type** – The type of the document.
- **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
- **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".
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type Valid choices: internal, external, external_gte, force

ping(***kwargs*)
Returns whether the cluster is running. https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html

put_script(***kwargs*)
Creates or updates a script. https://www.elastic.co/guide/en/elasticsearch/reference/master/modules-scripting.html

Parameters

- **id** – Script ID
- **body** – The document
- **context** – Script context
- **context** – Context name to compile script against
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit operation timeout

rank_eval(***kwargs*)
Allows to evaluate the quality of ranked search results over a set of typical search queries https://www.elastic.co/guide/en/elasticsearch/reference/master/search-rank-eval.html

Parameters

- **body** – The ranking evaluation search definition, including search requests, document ratings and ranking metric definition.
- **index** – A comma-separated list of index names to search; 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. Valid choices: open, closed, none, all Default: open
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

reindex(***kwargs*)
Allows to copy documents from one index to another, optionally filtering the source documents by a query, changing the destination index settings, or fetching the documents from a remote cluster. https://www.elastic.co/guide/en/elasticsearch/reference/master/docs-reindex.html

Parameters

- **body** – The search definition using the Query DSL and the prototype for the index request.
- **max_docs** – Maximum number of documents to process (default: all documents)
- **refresh** – Should the effected indexes be refreshed?
- **requests_per_second** – The throttle to set on this request in sub-requests per second. -1 means no throttle.
- **scroll** – Control how long to keep the search context alive Default: 5m
- **slices** – The number of slices this task should be divided into. Defaults to 1 meaning the task isn't sliced into subtasks. Default: 1
- **timeout** – Time each individual bulk request should wait for shards that are unavailable. Default: 1m

- **wait_for_active_shards** – Sets the number of shard copies that must be active before proceeding with the reindex operation. Defaults to 1, meaning the primary shard only. Set to *all* for all shard copies, otherwise set to any non-negative value less than or equal to the total number of copies for the shard (number of replicas + 1)
- **wait_for_completion** – Should the request should block until the reindex is complete. Default: True

reindex_rethrottle(***kwargs*)
Changes the number of requests per second for a particular Reindex operation. https://www.elastic.co/guide/en/elasticsearch/reference/master/docs-reindex.html

Parameters

- **task_id** – The task id to rethrottle
- **requests_per_second** – The throttle to set on this request in floating sub-requests per second. -1 means set no throttle.

render_search_template(***kwargs*)
Allows to use the Mustache language to pre-render a search definition. https://www.elastic.co/guide/en/elasticsearch/reference/current/search-template.html#_validating_templates

Parameters

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

scripts_painless_execute(***kwargs*)
Allows an arbitrary script to be executed and a result to be returned https://www.elastic.co/guide/en/elasticsearch/painless/master/painless-execute-api.html

Parameters **body** – The script to execute

scroll(***kwargs*)
Allows to retrieve a large numbers of results from a single search request. https://www.elastic.co/guide/en/elasticsearch/reference/master/search-request-body.html#request-body-search-scroll

Parameters

- **body** – The scroll ID if not passed by URL or query parameter.
- **scroll_id** – The scroll ID
- **rest_total_hits_as_int** – Indicates whether hits.total should be rendered as an integer or an object in the rest search response
- **scroll** – Specify how long a consistent view of the index should be maintained for scrolled search
- **scroll_id** – The scroll ID for scrolled search

search(***kwargs*)
Returns results matching a query. https://www.elastic.co/guide/en/elasticsearch/reference/master/search-search.html

Parameters

- **body** – The search definition using the Query DSL
- **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
- **_source** – True or false to return the _source field or not, or a list of fields to return
- **_source_excludes** – A list of fields to exclude from the returned _source field
- **_source_includes** – 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)
- **allow_partial_search_results** – Indicate if an error should be returned if there is a partial search failure or timeout Default: True
- **analyze_wildcard** – Specify whether wildcard and prefix queries should be analyzed (default: false)
- **analyzer** – The analyzer to use for the query string
- **batched_reduce_size** – The number of shard results that should be reduced at once on the coordinating node. This value should be used as a protection mechanism to reduce the memory overhead per search request if the potential number of shards in the request can be large. Default: 512
- **ccs_minimize_roundtrips** – Indicates whether network round- trips should be minimized as part of cross-cluster search requests execution Default: true
- **default_operator** – The default operator for query string query (AND or OR) Valid choices: AND, OR Default: OR
- **df** – The field to use as default where no field prefix is given in the query string
- **docvalue_fields** – A comma-separated list of fields to return as the docvalue representation of a field for each hit
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both. Valid choices: open, closed, none, all Default: open
- **explain** – Specify whether to return detailed information about score computation as part of a hit
- **from** – Starting offset (default: 0)
- **ignore_throttled** – Whether specified concrete, expanded or aliased indices should be ignored when throttled
- **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
- **max_concurrent_shard_requests** – The number of concurrent shard requests per node this search executes concurrently. This value should be used to limit the impact of the search on the cluster in order to limit the number of concurrent shard requests Default: 5
- **pre_filter_shard_size** – A threshold that enforces a pre- filter roundtrip to prefilter search shards based on query rewriting if the number of shards the search request expands to exceeds the threshold. This filter roundtrip can limit the number of shards significantly if for instance a shard can not match any documents based on it's rewrite method ie. if date filters are mandatory to match but the shard bounds and the query are disjoint. Default: 128

- **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
- **rest_total_hits_as_int** – Indicates whether hits.total should be rendered as an integer or an object in the rest search response
- **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: query_then_fetch, dfs_query_then_fetch
- **seq_no_primary_term** – Specify whether to return sequence number and primary term of the last modification of each hit
- **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
- **stored_fields** – A comma-separated list of stored fields to return as part of a hit
- **suggest_field** – Specify which field to use for suggestions
- **suggest_mode** – Specify suggest mode Valid choices: missing, popular, always Default: missing
- **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
- **track_total_hits** – Indicate if the number of documents that match the query should be tracked
- **typed_keys** – Specify whether aggregation and suggester names should be prefixed by their respective types in the response
- **version** – Specify whether to return document version as part of a hit

search_shards(***kwargs*)

Returns information about the indices and shards that a search request would be executed against. https://www.elastic.co/guide/en/elasticsearch/reference/master/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
- **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. Valid choices: open, closed, none, all Default: open
- **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(***kwargs*)

Allows to use the Mustache language to pre-render a search definition. https://www.elastic.co/guide/en/elasticsearch/reference/current/search-template.html

Parameters

- **body** – The search definition template and its params
- **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. Valid choices: open, closed, none, all Default: open
- **explain** – Specify whether to return detailed information about score computation as part of a hit
- **ignore_throttled** – Whether specified concrete, expanded or aliased indices should be ignored when throttled
- **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)
- **profile** – Specify whether to profile the query execution
- **rest_total_hits_as_int** – Indicates whether hits.total should be rendered as an integer or an object in the rest search response
- **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: query_then_fetch, query_and_fetch, dfs_query_then_fetch, dfs_query_and_fetch
- **typed_keys** – Specify whether aggregation and suggester names should be prefixed by their respective types in the response

termvectors(***kwargs*)

Returns information and statistics about terms in the fields of a particular document. https://www.elastic.co/guide/en/elasticsearch/reference/master/docs-termvectors.html

Parameters

- **index** – The index in which the document resides.
- **body** – Define parameters and or supply a document to get termvectors for. See documentation.
- **doc_type** – The type of the document.
- **id** – The id of the document, when not specified a doc param should be supplied.
- **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
- **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.
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type Valid choices: internal, external, external_gte, force

update(***kwargs*)

Updates a document with a script or partial document. https://www.elastic.co/guide/en/elasticsearch/reference/master/docs-update.html

Parameters

- **index** – The name of the index
- **id** – Document ID
- **body** – The request definition requires either *script* or partial *doc*
- **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_excludes** – A list of fields to exclude from the returned _source field
- **_source_includes** – A list of fields to extract and return from the _source field
- **if_primary_term** – only perform the update operation if the last operation that has changed the document has the specified primary term
- **if_seq_no** – only perform the update operation if the last operation that has changed the document has the specified sequence number
- **lang** – The script language (default: painless)
- **refresh** – If *true* then refresh the effected shards to make this operation visible to search, if *wait_for* then wait for a refresh to make this operation visible to search, if *false* (the default) then do nothing with refreshes. Valid choices: true, false, wait_for

- **retry_on_conflict** – Specify how many times should the operation be retried when a conflict occurs (default: 0)
- **routing** – Specific routing value
- **timeout** – Explicit operation timeout
- **wait_for_active_shards** – Sets the number of shard copies that must be active before proceeding with the update operation. Defaults to 1, meaning the primary shard only. Set to *all* for all shard copies, otherwise set to any non-negative value less than or equal to the total number of copies for the shard (number of replicas + 1)

update_by_query(***kwargs*)

Performs an update on every document in the index without changing the source, for example to pick up a mapping change. https://www.elastic.co/guide/en/elasticsearch/reference/master/docs-update-by-query.html

Parameters

- **index** – A comma-separated list of index names to search; use *_all* or empty string to perform the operation on all indices
- **body** – The search definition using the Query DSL
- **doc_type** – A comma-separated list of document types to search; leave empty to perform the operation on all types
- **_source** – True or false to return the _source field or not, or a list of fields to return
- **_source_excludes** – A list of fields to exclude from the returned _source field
- **_source_includes** – 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
- **conflicts** – What to do when the update by query hits version conflicts? Valid choices: abort, proceed Default: abort
- **default_operator** – The default operator for query string query (AND or OR) Valid choices: AND, OR Default: 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. Valid choices: open, closed, none, all Default: open
- **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
- **max_docs** – Maximum number of documents to process (default: all documents)
- **pipeline** – Ingest pipeline to set on index requests made by this action. (default: none)
- **preference** – Specify the node or shard the operation should be performed on (default: random)

- **q** – Query in the Lucene query string syntax
- **refresh** – Should the effected indexes be refreshed?
- **request_cache** – Specify if request cache should be used for this request or not, defaults to index level setting
- **requests_per_second** – The throttle to set on this request in sub-requests per second. -1 means no throttle.
- **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
- **scroll_size** – Size on the scroll request powering the update by query
- **search_timeout** – Explicit timeout for each search request. Defaults to no timeout.
- **search_type** – Search operation type Valid choices: query_then_fetch, dfs_query_then_fetch
- **size** – Deprecated, please use *max_docs* instead
- **slices** – The number of slices this task should be divided into. Defaults to 1 meaning the task isn't sliced into subtasks. Default: 1
- **sort** – A comma-separated list of <field>:<direction> pairs
- **stats** – Specific 'tag' of the request for logging and statistical purposes
- **terminate_after** – The maximum number of documents to collect for each shard, upon reaching which the query execution will terminate early.
- **timeout** – Time each individual bulk request should wait for shards that are unavailable. Default: 1m
- **version** – Specify whether to return document version as part of a hit
- **version_type** – Should the document increment the version number (internal) on hit or not (reindex)
- **wait_for_active_shards** – Sets the number of shard copies that must be active before proceeding with the update by query operation. Defaults to 1, meaning the primary shard only. Set to *all* for all shard copies, otherwise set to any non-negative value less than or equal to the total number of copies for the shard (number of replicas + 1)
- **wait_for_completion** – Should the request should block until the update by query operation is complete. Default: True

update_by_query_rethrottle(***kwargs*)

Changes the number of requests per second for a particular Update By Query operation. https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-update-by-query.html

Parameters

- **task_id** – The task id to rethrottle
- **requests_per_second** – The throttle to set on this request in floating sub-requests per second. -1 means set no throttle.

7.1.3 Indices

class elasticsearch.client.**IndicesClient**(*client*)

analyze(***kwargs*)
Performs the analysis process on a text and return the tokens breakdown of the text. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-analyze.html

Parameters

- **body** – Define analyzer/tokenizer parameters and the text on which the analysis should be performed
- **index** – The name of the index to scope the operation
- **index** – The name of the index to scope the operation

clear_cache(***kwargs*)
Clears all or specific caches for one or more indices. https://www.elastic.co/guide/en/elasticsearch/reference/master/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. Valid choices: open, closed, none, all Default: open
- **fielddata** – Clear field data
- **fields** – A comma-separated list of fields to clear when using the *fielddata* parameter (default: all)
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **index** – A comma-separated list of index name to limit the operation
- **query** – Clear query caches
- **request** – Clear request cache

clone(***kwargs*)
Clones an index https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-clone-index.html

Parameters

- **index** – The name of the source index to clone
- **target** – The name of the target index to clone into
- **body** – The configuration for the target index (*settings* and *aliases*)
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit operation timeout
- **wait_for_active_shards** – Set the number of active shards to wait for on the cloned index before the operation returns.

close(***kwargs*)
Closes an index. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-open-close.html

Parameters

- **index** – A comma separated list of indices to close
- **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. Valid choices: open, closed, none, all Default: open
- **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
- **wait_for_active_shards** – Sets the number of active shards to wait for before the operation returns.

create(***kwargs*)

Creates an index with optional settings and mappings. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-create-index.html

Parameters

- **index** – The name of the index
- **body** – The configuration for the index (*settings* and *mappings*)
- **include_type_name** – Whether a type should be expected in the body of the mappings.
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit operation timeout
- **wait_for_active_shards** – Set the number of active shards to wait for before the operation returns.

delete(***kwargs*)

Deletes an index. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-delete-index.html

Parameters

- **index** – A comma-separated list of indices to delete; use *_all* or * string to delete all indices
- **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) Valid choices: open, closed, none, all Default: open
- **ignore_unavailable** – Ignore unavailable indexes (default: false)
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit operation timeout

delete_alias(***kwargs*)

Deletes an alias. https://www.elastic.co/guide/en/elasticsearch/reference/master/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(***kwargs*)
Deletes an index template. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-templates.html

Parameters

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

exists(***kwargs*)
Returns information about whether a particular index exists. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-exists.html

Parameters

- **index** – A comma-separated list of index names
- **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) Valid choices: open, closed, none, all Default: open
- **flat_settings** – Return settings in flat format (default: false)
- **ignore_unavailable** – Ignore unavailable indexes (default: false)
- **include_defaults** – Whether to return all default setting for each of the indices.
- **local** – Return local information, do not retrieve the state from master node (default: false)

exists_alias(***kwargs*)
Returns information about whether a particular alias exists. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-aliases.html

Parameters

- **name** – A comma-separated list of alias names to return
- **index** – A comma-separated list of index names to filter aliases
- **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. Valid choices: open, closed, none, all Default: 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(***kwargs*)
Returns information about whether a particular index template exists. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-templates.html

Parameters

- **name** – The comma separated names of the index templates
- **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

exists_type(***kwargs*)
Returns information about whether a particular document type exists. (DEPRECATED) https://www.elastic.co/guide/en/elasticsearch/reference/master/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. Valid choices: open, closed, none, all Default: open
- **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(***kwargs*)
Performs the flush operation on one or more indices. https://www.elastic.co/guide/en/elasticsearch/reference/master/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. Valid choices: open, closed, none, all Default: open
- **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 true. If set to false the flush will be skipped iff if another flush operation is already running.

flush_synced(***kwargs*)
Performs a synced flush operation on one or more indices. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-synced-flush-api.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. Valid choices: open, closed, none, all Default: open
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

forcemerge(***kwargs*)
Performs the force merge operation on one or more indices. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-forcemerge.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. Valid choices: open, closed, none, all Default: open
- **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

freeze(***kwargs*)
https://www.elastic.co/guide/en/elasticsearch/reference/current/frozen.html

Parameters

- **index** – The name of the index to freeze
- **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. Valid choices: open, closed, none, all Default: closed
- **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

- **wait_for_active_shards** – Sets the number of active shards to wait for before the operation returns.

get(***kwargs*)
Returns information about one or more indices. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-get-index.html

Parameters

- **index** – A comma-separated list of index names
- **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) Valid choices: open, closed, none, all Default: open
- **flat_settings** – Return settings in flat format (default: false)
- **ignore_unavailable** – Ignore unavailable indexes (default: false)
- **include_defaults** – Whether to return all default setting for each of the indices.
- **include_type_name** – Whether to add the type name to the response (default: false)
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Specify timeout for connection to master

get_alias(***kwargs*)
Returns an alias. https://www.elastic.co/guide/en/elasticsearch/reference/master/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. Valid choices: open, closed, none, all Default: 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_field_mapping(***kwargs*)
Returns mapping for one or more fields. https://www.elastic.co/guide/en/elasticsearch/reference/master/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. Valid choices: open, closed, none, all Default: open
- **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
- **include_type_name** – Whether a type should be returned in the body of the mappings.
- **local** – Return local information, do not retrieve the state from master node (default: false)

get_mapping(***kwargs*)

Returns mappings for one or more indices. https://www.elastic.co/guide/en/elasticsearch/reference/master/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. Valid choices: open, closed, none, all Default: open
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **include_type_name** – Whether to add the type name to the response (default: false)
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Specify timeout for connection to master

get_settings(***kwargs*)

Returns settings for one or more indices. https://www.elastic.co/guide/en/elasticsearch/reference/master/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. Valid choices: open, closed, none, all Default: ['open', 'closed']
- **flat_settings** – Return settings in flat format (default: false)
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **include_defaults** – Whether to return all default setting for each of the indices.
- **local** – Return local information, do not retrieve the state from master node (default: false)

- **master_timeout** – Specify timeout for connection to master

get_template (***kwargs*)
Returns an index template. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-templates.html

Parameters

- **name** – The comma separated names of the index templates
- **flat_settings** – Return settings in flat format (default: false)
- **include_type_name** – Whether a type should be returned in the body of the mappings.
- **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 (***kwargs*)
The _upgrade API is no longer useful and will be removed. https://www.elastic.co/guide/en/elasticsearch/reference/master/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. Valid choices: open, closed, none, all Default: open
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

open (***kwargs*)
Opens an index. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-open-close.html

Parameters

- **index** – A comma separated list of indices to open
- **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. Valid choices: open, closed, none, all Default: closed
- **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
- **wait_for_active_shards** – Sets the number of active shards to wait for before the operation returns.

put_alias (***kwargs*)
Creates or updates an alias. https://www.elastic.co/guide/en/elasticsearch/reference/master/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(***kwargs*)
Updates the index mappings. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-put-mapping.html

Parameters

- **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.
- **doc_type** – The name of the document type
- **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. Valid choices: open, closed, none, all Default: open
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **include_type_name** – Whether a type should be expected in the body of the mappings.
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit operation timeout

put_settings(***kwargs*)
Updates the index settings. https://www.elastic.co/guide/en/elasticsearch/reference/master/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. Valid choices: open, closed, none, all Default: open
- **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
- **preserve_existing** – Whether to update existing settings. If set to *true* existing settings on an index remain unchanged, the default is *false*

- **timeout** – Explicit operation timeout

put_template(***kwargs*)
Creates or updates an index template. https://www.elastic.co/guide/en/elasticsearch/reference/master/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
- **flat_settings** – Return settings in flat format (default: false)
- **include_type_name** – Whether a type should be returned in the body of the mappings.
- **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

recovery(***kwargs*)
Returns information about ongoing index shard recoveries. https://www.elastic.co/guide/en/elasticsearch/reference/master/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
- **detailed** – Whether to display detailed information about shard recovery

refresh(***kwargs*)
Performs the refresh operation in one or more indices. https://www.elastic.co/guide/en/elasticsearch/reference/master/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. Valid choices: open, closed, none, all Default: open
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

reload_search_analyzers(***kwargs*)
https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-reload-analyzers.html

Parameters

- **index** – A comma-separated list of index names to reload analyzers for

- **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. Valid choices: open, closed, none, all Default: open
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

rollover(***kwargs*)

Updates an alias to point to a new index when the existing index is considered to be too large or too old. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-rollover-index.html

Parameters

- **alias** – The name of the alias to rollover
- **body** – The conditions that needs to be met for executing rollover
- **new_index** – The name of the rollover index
- **dry_run** – If set to true the rollover action will only be validated but not actually performed even if a condition matches. The default is false
- **include_type_name** – Whether a type should be included in the body of the mappings.
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit operation timeout
- **wait_for_active_shards** – Set the number of active shards to wait for on the newly created rollover index before the operation returns.

segments(***kwargs*)

Provides low-level information about segments in a Lucene index. https://www.elastic.co/guide/en/elasticsearch/reference/master/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. Valid choices: open, closed, none, all Default: open
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **verbose** – Includes detailed memory usage by Lucene.

shard_stores(***kwargs*)

Provides store information for shard copies of indices. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-shards-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. Valid choices: open, closed, none, all Default: open
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **status** – A comma-separated list of statuses used to filter on shards to get store information for Valid choices: green, yellow, red, all

shrink(***kwargs*)

Allow to shrink an existing index into a new index with fewer primary shards. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-shrink-index.html

Parameters

- **index** – The name of the source index to shrink
- **target** – The name of the target index to shrink into
- **body** – The configuration for the target index (*settings* and *aliases*)
- **copy_settings** – whether or not to copy settings from the source index (defaults to false)
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit operation timeout
- **wait_for_active_shards** – Set the number of active shards to wait for on the shrunken index before the operation returns.

split(***kwargs*)

Allows you to split an existing index into a new index with more primary shards. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-split-index.html

Parameters

- **index** – The name of the source index to split
- **target** – The name of the target index to split into
- **body** – The configuration for the target index (*settings* and *aliases*)
- **copy_settings** – whether or not to copy settings from the source index (defaults to false)
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit operation timeout
- **wait_for_active_shards** – Set the number of active shards to wait for on the shrunken index before the operation returns.

stats(***kwargs*)

Provides statistics on operations happening in an index. https://www.elastic.co/guide/en/elasticsearch/reference/master/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. Valid choices: _all, completion, docs, fielddata, query_cache, flush, get, indexing, merge, request_cache, refresh, search, segments, store, warmer, suggest

- **completion_fields** – A comma-separated list of fields for *fielddata* and *suggest* index metric (supports wildcards)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both. Valid choices: open, closed, none, all Default: open
- **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)
- **forbid_closed_indices** – If set to false stats will also collected from closed indices if explicitly specified or if expand_wildcards expands to closed indices Default: True
- **groups** – A comma-separated list of search groups for *search* index metric
- **include_segment_file_sizes** – Whether to report the aggregated disk usage of each one of the Lucene index files (only applies if segment stats are requested)
- **include_unloaded_segments** – If set to true segment stats will include stats for segments that are not currently loaded into memory
- **level** – Return stats aggregated at cluster, index or shard level Valid choices: cluster, indices, shards Default: indices
- **types** – A comma-separated list of document types for the *indexing* index metric

unfreeze(***kwargs*)
https://www.elastic.co/guide/en/elasticsearch/reference/current/frozen.html

Parameters

- **index** – The name of the index to unfreeze
- **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. Valid choices: open, closed, none, all Default: closed
- **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
- **wait_for_active_shards** – Sets the number of active shards to wait for before the operation returns.

update_aliases(***kwargs*)
Updates index aliases. https://www.elastic.co/guide/en/elasticsearch/reference/master/indices-aliases.html

Parameters

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

upgrade(***kwargs*)
The _upgrade API is no longer useful and will be removed. https://www.elastic.co/guide/en/elasticsearch/reference/master/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. Valid choices: open, closed, none, all Default: open
- **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(***kwargs*)

Allows a user to validate a potentially expensive query without executing it. https://www.elastic.co/guide/en/elasticsearch/reference/master/search-validate.html

Parameters

- **body** – The query definition specified with the Query DSL
- **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
- **all_shards** – Execute validation on all shards instead of one random shard per 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)
- **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) Valid choices: AND, OR Default: 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. Valid choices: open, closed, none, all Default: open
- **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
- **q** – Query in the Lucene query string syntax
- **rewrite** – Provide a more detailed explanation showing the actual Lucene query that will be executed.

7.1.4 Ingest

class elasticsearch.client.**IngestClient**(*client*)

delete_pipeline(***kwargs*)
Deletes a pipeline. https://www.elastic.co/guide/en/elasticsearch/reference/master/delete-pipeline-api.html

Parameters

- **id** – Pipeline ID
- **master_timeout** – Explicit operation timeout for connection to master node
- **timeout** – Explicit operation timeout

get_pipeline(***kwargs*)
Returns a pipeline. https://www.elastic.co/guide/en/elasticsearch/reference/master/get-pipeline-api.html

Parameters

- **id** – Comma separated list of pipeline ids. Wildcards supported
- **master_timeout** – Explicit operation timeout for connection to master node

processor_grok(***kwargs*)
Returns a list of the built-in patterns. https://www.elastic.co/guide/en/elasticsearch/reference/master/grok-processor.html#grok-processor-rest-get

put_pipeline(***kwargs*)
Creates or updates a pipeline. https://www.elastic.co/guide/en/elasticsearch/reference/master/put-pipeline-api.html

Parameters

- **id** – Pipeline ID
- **body** – The ingest definition
- **master_timeout** – Explicit operation timeout for connection to master node
- **timeout** – Explicit operation timeout

simulate(***kwargs*)
Allows to simulate a pipeline with example documents. https://www.elastic.co/guide/en/elasticsearch/reference/master/simulate-pipeline-api.html

Parameters

- **body** – The simulate definition
- **id** – Pipeline ID
- **verbose** – Verbose mode. Display data output for each processor in executed pipeline

7.1.5 Cluster

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

allocation_explain(***kwargs*)
Provides explanations for shard allocations in the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/master/cluster-allocation-explain.html

Parameters

- **body** – The index, shard, and primary flag to explain. Empty means 'explain the first unassigned shard'
- **include_disk_info** – Return information about disk usage and shard sizes (default: false)
- **include_yes_decisions** – Return 'YES' decisions in explanation (default: false)

get_settings(***kwargs*)
Returns cluster settings. https://www.elastic.co/guide/en/elasticsearch/reference/master/cluster-update-settings.html

Parameters

- **flat_settings** – Return settings in flat format (default: false)
- **include_defaults** – Whether to return all default clusters setting.
- **master_timeout** – Explicit operation timeout for connection to master node
- **timeout** – Explicit operation timeout

health(***kwargs*)
Returns basic information about the health of the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/master/cluster-health.html

Parameters

- **index** – Limit the information returned to a specific index
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both. Valid choices: open, closed, none, all Default: all
- **level** – Specify the level of detail for returned information Valid choices: cluster, indices, shards Default: cluster
- **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_events** – Wait until all currently queued events with the given priority are processed Valid choices: immediate, urgent, high, normal, low, languid
- **wait_for_no_initializing_shards** – Whether to wait until there are no initializing shards in the cluster
- **wait_for_no_relocating_shards** – Whether to wait until there are no relocating shards in the cluster
- **wait_for_nodes** – Wait until the specified number of nodes is available
- **wait_for_status** – Wait until cluster is in a specific state Valid choices: green, yellow, red

pending_tasks(***kwargs*)
Returns a list of any cluster-level changes (e.g. create index, update mapping, allocate or fail shard) which have not yet been executed. https://www.elastic.co/guide/en/elasticsearch/reference/master/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(***kwargs*)
Updates the cluster settings. https://www.elastic.co/guide/en/elasticsearch/reference/master/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

remote_info(***kwargs*)
Returns the information about configured remote clusters. https://www.elastic.co/guide/en/elasticsearch/reference/master/cluster-remote-info.html

reroute(***kwargs*)
Allows to manually change the allocation of individual shards in the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/master/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: _all, blocks, metadata, nodes, routing_table, master_node, version
- **retry_failed** – Retries allocation of shards that are blocked due to too many subsequent allocation failures
- **timeout** – Explicit operation timeout

state(***kwargs*)
Returns a comprehensive information about the state of the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/master/cluster-state.html

Parameters

- **metric** – Limit the information returned to the specified metrics Valid choices: _all, blocks, metadata, nodes, routing_table, routing_nodes, master_node, version
- **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. Valid choices: open, closed, none, all Default: open

- **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
- **wait_for_metadata_version** – Wait for the metadata version to be equal or greater than the specified metadata version
- **wait_for_timeout** – The maximum time to wait for wait_for_metadata_version before timing out

stats(***kwargs*)
Returns high-level overview of cluster statistics. https://www.elastic.co/guide/en/elasticsearch/reference/master/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)
- **timeout** – Explicit operation timeout

7.1.6 Nodes

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

hot_threads(***kwargs*)
Returns information about hot threads on each node in the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/master/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: 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(***kwargs*)
Returns information about nodes in the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/master/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. Valid choices: settings, os, process, jvm, thread_pool, transport, http, plugins, ingest
- **flat_settings** – Return settings in flat format (default: false)
- **timeout** – Explicit operation timeout

reload_secure_settings(***kwargs*)
Reloads secure settings. https://www.elastic.co/guide/en/elasticsearch/reference/master/secure-settings.html#reloadable-secure-settings

Parameters

- **node_id** – A comma-separated list of node IDs to span the reload/reinit call. Should stay empty because reloading usually involves all cluster nodes.
- **timeout** – Explicit operation timeout

stats(***kwargs*)
Returns statistical information about nodes in the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/master/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 Valid choices: _all, breaker, fs, http, indices, jvm, os, process, thread_pool, transport, discovery
- **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. Valid choices: _all, completion, docs, fielddata, query_cache, flush, get, indexing, merge, request_cache, refresh, search, segments, store, warmer, suggest
- **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
- **include_segment_file_sizes** – Whether to report the aggregated disk usage of each one of the Lucene index files (only applies if segment stats are requested)
- **level** – Return indices stats aggregated at index, node or shard level Valid choices: indices, node, shards Default: node
- **timeout** – Explicit operation timeout
- **types** – A comma-separated list of document types for the *indexing* index metric

usage(***kwargs*)
Returns low-level information about REST actions usage on nodes. https://www.elastic.co/guide/en/elasticsearch/reference/master/cluster-nodes-usage.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 Valid choices: _all, rest_actions
- **timeout** – Explicit operation timeout

7.1.7 Cat

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

aliases(***kwargs*)

Shows information about currently configured aliases to indices including filter and routing infos. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat-alias.html

Parameters

- **name** – A comma-separated list of alias names to return
- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **s** – Comma-separated list of column names or column aliases to sort by
- **v** – Verbose mode. Display column headers

allocation(***kwargs*)

Provides a snapshot of how many shards are allocated to each data node and how much disk space they are using. https://www.elastic.co/guide/en/elasticsearch/reference/master/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: b, k, kb, m, mb, g, gb, t, tb, p, pb
- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **s** – Comma-separated list of column names or column aliases to sort by
- **v** – Verbose mode. Display column headers

count (***kwargs*)
Provides quick access to the document count of the entire cluster, or individual indices. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat-count.html

Parameters

- **index** – A comma-separated list of index names to limit the returned information
- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **s** – Comma-separated list of column names or column aliases to sort by
- **v** – Verbose mode. Display column headers

fielddata (***kwargs*)
Shows how much heap memory is currently being used by fielddata on every data node in the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/master/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: b, k, kb, m, mb, g, gb, t, tb, p, pb
- **fields** – A comma-separated list of fields to return in the output
- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **s** – Comma-separated list of column names or column aliases to sort by
- **v** – Verbose mode. Display column headers

health (***kwargs*)
Returns a concise representation of the cluster health. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat-health.html

Parameters

- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **s** – Comma-separated list of column names or column aliases to sort by
- **time** – The unit in which to display time values Valid choices: d (Days), h (Hours), m (Minutes), s (Seconds), ms (Milliseconds), micros (Microseconds), nanos (Nanoseconds)
- **ts** – Set to false to disable timestamping Default: True
- **v** – Verbose mode. Display column headers

help (***kwargs*)
Returns help for the Cat APIs. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat.html

Parameters

- **help** – Return help information

- **s** – Comma-separated list of column names or column aliases to sort by

indices(***kwargs*)
Returns information about indices: number of primaries and replicas, document counts, disk size, ... https://www.elastic.co/guide/en/elasticsearch/reference/master/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: b, k, m, g
- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **health** – A health status ("green", "yellow", or "red" to filter only indices matching the specified health status Valid choices: green, yellow, red
- **help** – Return help information
- **include_unloaded_segments** – If set to true segment stats will include stats for segments that are not currently loaded into memory
- **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
- **s** – Comma-separated list of column names or column aliases to sort by
- **time** – The unit in which to display time values Valid choices: d (Days), h (Hours), m (Minutes), s (Seconds), ms (Milliseconds), micros (Microseconds), nanos (Nanoseconds)
- **v** – Verbose mode. Display column headers

master(***kwargs*)
Returns information about the master node. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat-master.html

Parameters

- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **s** – Comma-separated list of column names or column aliases to sort by
- **v** – Verbose mode. Display column headers

nodeattrs(***kwargs*)
Returns information about custom node attributes. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat-nodeattrs.html

Parameters

- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display

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

nodes (***kwargs*)
Returns basic statistics about performance of cluster nodes. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat-nodes.html

Parameters

- **bytes** – The unit in which to display byte values Valid choices: b, k, kb, m, mb, g, gb, t, tb, p, pb
- **format** – a short version of the Accept header, e.g. json, yaml
- **full_id** – Return the full node ID instead of the shortened version (default: false)
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **s** – Comma-separated list of column names or column aliases to sort by
- **time** – The unit in which to display time values Valid choices: d (Days), h (Hours), m (Minutes), s (Seconds), ms (Milliseconds), micros (Microseconds), nanos (Nanoseconds)
- **v** – Verbose mode. Display column headers

pending_tasks (***kwargs*)
Returns a concise representation of the cluster pending tasks. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat-pending-tasks.html

Parameters

- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **s** – Comma-separated list of column names or column aliases to sort by
- **time** – The unit in which to display time values Valid choices: d (Days), h (Hours), m (Minutes), s (Seconds), ms (Milliseconds), micros (Microseconds), nanos (Nanoseconds)
- **v** – Verbose mode. Display column headers

plugins (***kwargs*)
Returns information about installed plugins across nodes node. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat-plugins.html

Parameters

- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **s** – Comma-separated list of column names or column aliases to sort by
- **v** – Verbose mode. Display column headers

recovery (***kwargs*)

Returns information about index shard recoveries, both on-going completed. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat-recovery.html

Parameters

- **index** – Comma-separated list or wildcard expression of index names to limit the returned information
- **active_only** – If *true*, the response only includes ongoing shard recoveries
- **bytes** – The unit in which to display byte values Valid choices: b, k, kb, m, mb, g, gb, t, tb, p, pb
- **detailed** – If *true*, the response includes detailed information about shard recoveries
- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **index** – Comma-separated list or wildcard expression of index names to limit the returned information
- **s** – Comma-separated list of column names or column aliases to sort by
- **time** – The unit in which to display time values Valid choices: d (Days), h (Hours), m (Minutes), s (Seconds), ms (Milliseconds), micros (Microseconds), nanos (Nanoseconds)
- **v** – Verbose mode. Display column headers

repositories (***kwargs*)

Returns information about snapshot repositories registered in the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat-repositories.html

Parameters

- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **local** – Return local information, do not retrieve the state from master node
- **master_timeout** – Explicit operation timeout for connection to master node
- **s** – Comma-separated list of column names or column aliases to sort by
- **v** – Verbose mode. Display column headers

segments (***kwargs*)
Provides low-level information about the segments in the shards of an index. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat-segments.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: b, k, kb, m, mb, g, gb, t, tb, p, pb
- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **s** – Comma-separated list of column names or column aliases to sort by
- **v** – Verbose mode. Display column headers

shards (***kwargs*)
Provides a detailed view of shard allocation on nodes. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat-shards.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: b, k, kb, m, mb, g, gb, t, tb, p, pb
- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **s** – Comma-separated list of column names or column aliases to sort by
- **time** – The unit in which to display time values Valid choices: d (Days), h (Hours), m (Minutes), s (Seconds), ms (Milliseconds), micros (Microseconds), nanos (Nanoseconds)
- **v** – Verbose mode. Display column headers

snapshots (***kwargs*)
Returns all snapshots in a specific repository. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat-snapshots.html

Parameters

- **repository** – Name of repository from which to fetch the snapshot information
- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **ignore_unavailable** – Set to true to ignore unavailable snapshots
- **master_timeout** – Explicit operation timeout for connection to master node

- **s** – Comma-separated list of column names or column aliases to sort by
- **time** – The unit in which to display time values Valid choices: d (Days), h (Hours), m (Minutes), s (Seconds), ms (Milliseconds), micros (Microseconds), nanos (Nanoseconds)
- **v** – Verbose mode. Display column headers

tasks(***kwargs*)
Returns information about the tasks currently executing on one or more nodes in the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/master/tasks.html

Parameters

- **actions** – A comma-separated list of actions that should be returned. Leave empty to return all.
- **detailed** – Return detailed task information (default: false)
- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **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
- **parent_task** – Return tasks with specified parent task id. Set to -1 to return all.
- **s** – Comma-separated list of column names or column aliases to sort by
- **time** – The unit in which to display time values Valid choices: d (Days), h (Hours), m (Minutes), s (Seconds), ms (Milliseconds), micros (Microseconds), nanos (Nanoseconds)
- **v** – Verbose mode. Display column headers

templates(***kwargs*)
Returns information about existing templates. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat-templates.html

Parameters

- **name** – A pattern that returned template names must match
- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **s** – Comma-separated list of column names or column aliases to sort by
- **v** – Verbose mode. Display column headers

thread_pool(***kwargs*)
Returns cluster-wide thread pool statistics per node. By default the active, queue and rejected statistics are returned for all thread pools. https://www.elastic.co/guide/en/elasticsearch/reference/master/cat-thread-pool.html

Parameters

- **thread_pool_patterns** – A comma-separated list of regular- expressions to filter the thread pools in the output
- **format** – a short version of the Accept header, e.g. json, yaml
- **h** – Comma-separated list of column names to display
- **help** – Return help information
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **s** – Comma-separated list of column names or column aliases to sort by
- **size** – The multiplier in which to display values Valid choices: , k, m, g, t, p
- **v** – Verbose mode. Display column headers

7.1.8 Snapshot

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

cleanup_repository(***kwargs*)
Removes stale data from repository. https://www.elastic.co/guide/en/elasticsearch/reference/master/modules-snapshots.html

Parameters

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

create(***kwargs*)
Creates a snapshot in a repository. https://www.elastic.co/guide/en/elasticsearch/reference/master/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

create_repository(***kwargs*)
Creates a repository. https://www.elastic.co/guide/en/elasticsearch/reference/master/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(***kwargs*)
Deletes a snapshot. https://www.elastic.co/guide/en/elasticsearch/reference/master/modules-snapshots.html

Parameters

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

delete_repository(***kwargs*)
Deletes a repository. https://www.elastic.co/guide/en/elasticsearch/reference/master/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(***kwargs*)
Returns information about a snapshot. https://www.elastic.co/guide/en/elasticsearch/reference/master/modules-snapshots.html

Parameters

- **repository** – A repository name
- **snapshot** – A comma-separated list of snapshot names
- **ignore_unavailable** – Whether to ignore unavailable snapshots, defaults to false which means a SnapshotMissingException is thrown
- **master_timeout** – Explicit operation timeout for connection to master node
- **verbose** – Whether to show verbose snapshot info or only show the basic info found in the repository index blob

get_repository(***kwargs*)
Returns information about a repository. https://www.elastic.co/guide/en/elasticsearch/reference/master/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(***kwargs*)
Restores a snapshot. https://www.elastic.co/guide/en/elasticsearch/reference/master/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

status(***kwargs*)
Returns information about the status of a snapshot. https://www.elastic.co/guide/en/elasticsearch/reference/master/modules-snapshots.html

Parameters

- **repository** – A repository name
- **snapshot** – A comma-separated list of snapshot names
- **ignore_unavailable** – Whether to ignore unavailable snapshots, defaults to false which means a SnapshotMissingException is thrown
- **master_timeout** – Explicit operation timeout for connection to master node

verify_repository(***kwargs*)
Verifies a repository. https://www.elastic.co/guide/en/elasticsearch/reference/master/modules-snapshots.html

Parameters

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

7.1.9 Tasks

class elasticsearch.client.**TasksClient**(*client*)

cancel(***kwargs*)
Cancels a task, if it can be cancelled through an API. https://www.elastic.co/guide/en/elasticsearch/reference/master/tasks.html

Parameters

- **task_id** – Cancel the task with specified task id (node_id:task_number)
- **actions** – A comma-separated list of actions that should be cancelled. Leave empty to cancel all.
- **nodes** – 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
- **parent_task_id** – Cancel tasks with specified parent task id (node_id:task_number). Set to -1 to cancel all.

get(***kwargs*)
Returns information about a task. https://www.elastic.co/guide/en/elasticsearch/reference/master/tasks.html

Parameters

- **task_id** – Return the task with specified id (node_id:task_number)
- **timeout** – Explicit operation timeout
- **wait_for_completion** – Wait for the matching tasks to complete (default: false)

list(***kwargs*)
Returns a list of tasks. https://www.elastic.co/guide/en/elasticsearch/reference/master/tasks.html

Parameters

- **actions** – A comma-separated list of actions that should be returned. Leave empty to return all.
- **detailed** – Return detailed task information (default: false)
- **group_by** – Group tasks by nodes or parent/child relationships Valid choices: nodes, parents, none Default: nodes
- **nodes** – 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
- **parent_task_id** – Return tasks with specified parent task id (node_id:task_number). Set to -1 to return all.
- **timeout** – Explicit operation timeout
- **wait_for_completion** – Wait for the matching tasks to complete (default: false)

7.2 X-Pack APIs

X-Pack is an Elastic Stack extension that bundles security, alerting, monitoring, reporting, and graph capabilities into one easy-to-install package. While the X-Pack components are designed to work together seamlessly, you can easily enable or disable the features you want to use.

7.2.1 Info

X-Pack info provides general info about the installed X-Pack.

class elasticsearch.client.xpack.**XPackClient**(*client*)

info(***kwargs*)
https://www.elastic.co/guide/en/elasticsearch/reference/current/info-api.html

Parameters **categories** – Comma-separated list of info categories. Can be any of: build, license, features

usage(***kwargs*)
Retrieveinformationaboutxpackfeaturesusage

Parameters **master_timeout** – Specify timeout for watch write operation

7.2.2 Graph Explore

X-Pack Graph Explore enables you to extract and summarize information about the documents and terms in your Elasticsearch index.

7.2.3 Licensing API

Licensing API can be used to manage your licences.

7.2.4 Machine Learning APIs

Machine Learning can be useful for discovering new patterns about your data. For a more detailed explanation about X-Pack's machine learning please refer to the official documentation.

7.2.5 Security APIs

Security API can be used to help secure your Elasticsearch cluster. Integrating with LDAP and Active Directory.

7.2.6 Watcher APIs

Watcher API can be used to notify you when certain pre-defined thresholds have happened.

7.2.7 Migration APIs

Migration API helps simplify upgrading X-Pack indices from one version to another.

7.3 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, underlying exception when not.

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.**AuthenticationException**(*TransportError*)
Exception representing a 401 status code.

class elasticsearch.**AuthorizationException**(*TransportError*)
Exception representing a 403 status code.

7.4 Connection Layer API

All of the classes responsible for handling the connection to the Elasticsearch cluster. The default subclasses used can be overriden 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.

7.4.1 Transport

class elasticsearch.**Transport**(*hosts*, *connection_class=Urllib3HttpConnection*, *connection_pool_class=ConnectionPool*, *host_info_callback=construct_hosts_list*, *sniff_on_start=False*, *sniffer_timeout=None*, *sniff_on_connection_fail=False*, *serializer=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 `(502, 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

close()
Explicitly closes connections

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*, *headers=None*, *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 successful 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
- **headers** – dictionary of headers, will be handed over to the underlying `Connection` class

- **params** – dictionary of query parameters, will be handed over to the underlying `Connection` class for serialization
- **body** – body of the request, will be serialized using serializer and passed to the connection

set_connections(*hosts*)
Instantiate all the connections and create 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`

7.4.2 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 (its 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

close()
Explicitly closes connections

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 availible 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.

7.4.3 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

7.4.4 Urllib3HttpConnection (default connection_class)

If you have complex SSL logic for connecting to Elasticsearch using an *SSLContext* object might be more helpful. You can create one natively using the python SSL library with the *create_default_context* (https://docs.python.org/3/library/ssl.html#ssl.create_default_context) method.

To create an *SSLContext* object you only need to use one of cafile, capath or cadata:

```
>>> from ssl import create_default_context
>>> context = create_default_context(cafile=None, capath=None, cadata=None)
```

- *cafile* is the path to your CA File
- *capath* is the directory of a collection of CA's
- *cadata* is either an ASCII string of one or more PEM-encoded certificates or a bytes-like object of DER-encoded certificates.

Please note that the use of SSLContext is only available for Urllib3.

class elasticsearch.**Urllib3HttpConnection**(*host='localhost'*, *port=9200*, *http_auth=None*, *use_ssl=False*, *verify_certs=None*, *ssl_show_warn=True*, *ca_certs=None*, *client_cert=None*, *client_key=None*, *ssl_version=None*, *ssl_assert_hostname=None*, *ssl_assert_fingerprint=None*, *maxsize=10*, *headers=None*, *ssl_context=None*, *http_compress=False*, *cloud_id=None*, *api_key=None*, ***kwargs*)

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

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)
- **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
- **ssl_show_warn** – show warning when verify certs is disabled
- **ca_certs** – optional path to CA bundle. See https://urllib3.readthedocs.io/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, or cert only if using client_key
- **client_key** – path to the file containing the private key if using separate cert and key files (client_cert will contain only the cert)
- **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).
- **ssl_assert_hostname** – use hostname verification if not *False*
- **ssl_assert_fingerprint** – verify the supplied certificate fingerprint if not *None*
- **maxsize** – the number of connections which will be kept open to this host. See https://urllib3.readthedocs.io/en/1.4/pools.html#api for more information.
- **headers** – any custom http headers to be add to requests
- **http_compress** – Use gzip compression
- **cloud_id** – The Cloud ID from ElasticCloud. Convient way to connect to cloud instances.
- **api_key** – optional API Key authentication as either base64 encoded string or a tuple. Other host connection params will be ignored.

close()

Explicitly closes connection

7.5 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)
```

The default connection class is based on `urllib3` which is more performant and lightweight than the optional `requests`-based class. Only use `RequestsHttpConnection` if you have need of any of `requests` advanced features like custom auth plugins etc.

7.5.1 Connection

class elasticsearch.connection.**Connection**(*host='localhost'*, *port=9200*, *use_ssl=False*, *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)

7.5.2 Urllib3HttpConnection

class elasticsearch.connection.**Urllib3HttpConnection**(*host='localhost'*, *port=9200*, *http_auth=None*, *use_ssl=False*, *verify_certs=None*, *ssl_show_warn=True*, *ca_certs=None*, *client_cert=None*, *client_key=None*, *ssl_version=None*, *ssl_assert_hostname=None*, *ssl_assert_fingerprint=None*, *maxsize=10*, *headers=None*, *ssl_context=None*, *http_compress=False*, *cloud_id=None*, *api_key=None*, ***kwargs*)

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

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)
- **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
- **ssl_show_warn** – show warning when verify certs is disabled
- **ca_certs** – optional path to CA bundle. See https://urllib3.readthedocs.io/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, or cert only if using client_key
- **client_key** – path to the file containing the private key if using separate cert and key files (client_cert will contain only the cert)
- **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).
- **ssl_assert_hostname** – use hostname verification if not *False*
- **ssl_assert_fingerprint** – verify the supplied certificate fingerprint if not *None*
- **maxsize** – the number of connections which will be kept open to this host. See https://urllib3.readthedocs.io/en/1.4/pools.html#api for more information.
- **headers** – any custom http headers to be add to requests
- **http_compress** – Use gzip compression
- **cloud_id** – The Cloud ID from ElasticCloud. Convient way to connect to cloud instances.
- **api_key** – optional API Key authentication as either base64 encoded string or a tuple. Other host connection params will be ignored.

7.5.3 RequestsHttpConnection

class elasticsearch.connection.**RequestsHttpConnection**(*host='localhost'*, *port=9200*, *http_auth=None*, *use_ssl=False*, *verify_certs=True*, *ssl_show_warn=True*, *ca_certs=None*, *client_cert=None*, *client_key=None*, *headers=None*, *cloud_id=None*, *api_key=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
- **ssl_show_warn** – show warning when verify certs is disabled
- **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, or cert only if using client_key
- **client_key** – path to the file containing the private key if using separate cert and key files (client_cert will contain only the cert)
- **headers** – any custom http headers to be add to requests
- **cloud_id** – The Cloud ID from ElasticCloud. Convient way to connect to cloud instances.
- **api_key** – optional API Key authentication as either base64 encoded string or a tuple. Other host connection params will be ignored.

7.6 Helpers

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

7.6.1 Bulk helpers

There are several helpers for the `bulk` API since its 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,
    '_routing': 5,
    'pipeline': 'my-ingest-pipeline',
    '_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,
    "_routing": 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.'}
}
```

Example:

Lets say we have an iterable of data. Lets say a list of words called `mywords` and we want to index those words into individual documents where the structure of the document is like `{"word": "<myword>"}`.

```
def gendata():
    mywords = ['foo', 'bar', 'baz']
    for word in mywords:
        yield {
            "_index": "mywords",
            "_type": "document",
            "doc": {"word": word},
        }

bulk(es, gendata())
```

For a more complete and complex example please take a look at https://github.com/elastic/elasticsearch-py/blob/master/example/load.py#L76-L130

The `parallel_bulk()` api is a wrapper around the *bulk()* api to provide threading. `parallel_bulk()` returns a generator which must be consumed to produce results.

To see the results use:

```
for success, info in parallel_bulk(...):
if not success:
    print('A document failed:', info)
```

If you don't care about the results, you can use deque from collections:

```
from collections import deque
deque(parallel_bulk(...), maxlen=0)
```

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=104857600*, *raise_on_error=True*, *expand_action_callback=<function expand_action>*, *raise_on_exception=True*, *max_retries=0*, *initial_backoff=2*, *max_backoff=600*, *yield_ok=True*, **args*, ***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.

If you specify `max_retries` it will also retry any documents that were rejected with a `429` status code. To do this it will wait (**by calling time.sleep which will block**) for `initial_backoff` seconds and then, every subsequent rejection for the same chunk, for double the time every time up to `max_backoff` seconds.

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).
- **max_retries** – maximum number of times a document will be retried when `429` is received, set to 0 (default) for no retries on `429`
- **initial_backoff** – number of seconds we should wait before the first retry. Any subsequent retries will be powers of `initial_backoff * 2**retry_number`
- **max_backoff** – maximum number of seconds a retry will wait
- **yield_ok** – if set to False will skip successful documents in the output

elasticsearch.helpers.**parallel_bulk**(*client*, *actions*, *thread_count=4*, *chunk_size=500*, *max_chunk_bytes=104857600*, *queue_size=4*, *expand_action_callback=<function expand_action>*, **args*, ***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).
- **queue_size** – size of the task queue between the main thread (producing chunks to send) and the processing threads.

`elasticsearch.helpers.`**`bulk`**(*client*, *actions*, *stats_only=False*, **args*, ***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`. Note that by default we raise a `BulkIndexError` when we encounter an error so options like `stats_only` only apply when `raise_on_error` is set to `False`.

When errors are being collected original document data is included in the error dictionary which can lead to an extra high memory usage. If you need to process a lot of data and want to ignore/collect errors please consider using the *streaming_bulk()* helper which will just return the errors and not store them in memory.

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, see *streaming_bulk()* for more accepted parameters.

7.6.2 Scan

`elasticsearch.helpers.`**`scan`**(*client*, *query=None*, *scroll='5m'*, *raise_on_error=True*, *preserve_order=False*, *size=1000*, *request_timeout=None*, *clear_scroll=True*, *scroll_kwargs=None*, ***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.
- **size** – size (per shard) of the batch send at each iteration.

- **request_timeout** – explicit timeout for each call to `scan`
- **clear_scroll** – explicitly calls delete on the scroll id via the clear scroll API at the end of the method on completion or error, defaults to true.
- **scroll_kwargs** – additional kwargs to be passed to *scroll()*

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

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

7.6.3 Reindex

elasticsearch.helpers.**reindex**(*client*, *source_index*, *target_index*, *query=None*, *target_client=None*, *chunk_size=500*, *scroll='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.

Since `2.3` a *reindex()* api is available as part of elasticsearch itself. It is recommended to use the api instead of this helper wherever possible. The helper is here mostly for backwards compatibility and for situations where more flexibility is needed.

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()*

7.7 Changelog

7.7.1 7.5.1 (2020-01-19)

- `7.5.0` tag was not released so retagging

7.7.2 7.5.0

- All API is now auto generated
- deprecated the `.xpack` namespace
- Update client to support ES 7.5 APIs

7.7.3 7.1.0 (2019-11-14)

- Fix sniffing with `http.publish_host`
- Fix `request_timeout` for `indices` APIs
- Allow access to `x-pack` features without `xpack` namespace
- Fix mark dead

7.7.4 7.0.5 (2019-10-01)

- Fix `verify_certs=False`

7.7.5 7.0.4 (2019-08-22)

- Fix wheel distribution

7.7.6 7.0.3 (2019-08-21)

- remove sleep in retries
- pass `scroll_id` through body in `scroll`
- add `user-agent`

7.7.7 7.0.2 (2019-05-29)

- Add connection parameter for Elastic Cloud cloud_id.
- ML client uses client object for _bulk_body requests

7.7.8 7.0.1 (2019-05-19)

- Use black to format the code.
- Update the test matrix to only use current pythons and 7.x ES
- Blocking pool must fit thread_count
- Update client to support missing ES 7 API's and query params.

7.7.9 7.0.0 (2019-04-11)

- Removed deprecated option `update_all_types`.
- Using insecure SSL configuration (`verify_cert=False`) raises a warning, this can be not showed with `ssl_show_warn=False`
- Add support for 7.x api's in Elasticsearch both xpack and oss flavors

7.7.10 6.3.0 (2018-06-20)

- Add an exponential wait on delays
- Fix issues with dependencies
- Adding X-pack Docs
- Adding forecast to x-pack ML client

7.7.11 6.2.0 (2018-03-20)

- cleanup for SSL Context
- Add X-Pack clients to -py
- Adding Gzip support for capacity constrained networks
- `_routing` in bulk action has been deprecated in ES. Introduces a breaking change if you use `routing` as a field in your documents.

7.7.12 6.1.1 (2018-01-05)

- Updates to SSLContext logic to make it easier to use and have saner defaults.
- Doc updates

7.7.13 6.1.0 (2018-01-05)

- bad release

7.7.14 6.0.0 (2017-11-14)

- compatibility with Elasticsearch 6.0.0

7.7.15 5.5.0 (2017-11-10)

- `streaming_bulk` helper now supports retries with incremental backoff
- `scan` helper properly checks for successful shards instead of just checking `failed`
- compatible release with elasticsearch `5.6.4`
- fix handling of UTF-8 surrogates

7.7.16 5.4.0 (2017-05-18)

- `bulk` helpers now extract `pipeline` parameter from the action dictionary.

7.7.17 5.3.0 (2017-03-30)

Compatibility with elasticsearch 5.3

7.7.18 5.2.0 (2017-02-12)

The client now automatically sends `Content-Type` http header set to `application/json`. If you are explicitly passing in other encoding than `json` you need to set the header manually.

7.7.19 5.1.0 (2017-01-11)

- Fixed sniffing

7.7.20 5.0.1 (2016-11-02)

Fixed performance regression in `scan` helper

7.7.21 5.0.0 (2016-10-19)

Version compatible with elasticsearch 5.0

- when using SSL certificate validation is now on by default. Install `certifi` or supply root certificate bundle.
- `elasticsearch.trace` logger now also logs failed requests, signature of internal logging method `log_request_fail` has changed, all custom connection classes need to be updated
- added `headers` arg to connections to support custom http headers
- passing in a keyword parameter with `None` as value will cause that param to be ignored

7.7.22 2.4.0 (2016-08-17)

- `ping` now ignores all `TransportError` exceptions and just returns `False`
- expose `scroll_id` on `ScanError`
- increase default size for `scan` helper to 1000

Internal:

- changed `Transport.perform_request` to just return the body, not status as well.

7.7.23 2.3.0 (2016-02-29)

- added `client_key` argument to configure client certificates
- debug logging now includes response body even for failed requests

7.7.24 2.2.0 (2016-01-05)

Due to change in json encoding the client will no longer mask issues with encoding - if you work with non-ascii data in python 2 you must use the `unicode` type or have proper encoding set in your environment.

- adding additional options for ssh - `ssl_assert_hostname` and `ssl_assert_fingerprint` to the default connection class
- fix sniffing

7.7.25 2.1.0 (2015-10-19)

- move multiprocessing import inside parallel bulk for Google App Engine

7.7.26 2.0.0 (2015-10-14)

- Elasticsearch 2.0 compatibility release

7.7.27 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
- In helpers, removed `bulk_index` as an alias for `bulk`. Use `bulk` instead.

7.7.28 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

7.7.29 1.6.0 (2015-06-10)

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

7.7.30 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

7.7.31 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

7.7.32 1.3.0 (2014-12-31)

- Timeout now doesn't trigger a retry by default (can be overriden 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

7.7.33 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

7.7.34 1.1.1 (2014-07-04)

Bugfix release fixing escaping issues with `request_timeout`.

7.7.35 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

7.7.36 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.

7.7.37 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 `incices.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

7.7.38 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

7.7.39 0.4.2 (2013-10-08)

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

7.7.40 0.4.1 (2013-09-24)

Initial release.

CHAPTER 8

License

CHAPTER 9

Indices and tables

- genindex
- modindex
- search

Python Module Index

e

Index

F

G

H

I

L

M

T

U

V

X