Elasticsearch Documentation

Release 7.9.0

Honza Král

Aug 18, 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.

CHAPTER 1

Installation

Install the `elasticsearch` package with pip:

```
$ python -m pip install elasticsearch
```

If your application uses async/await in Python you can install with the `async` extra:

```
$ python -m pip install elasticsearch[async]
```

Read more about how to use asyncio with this project.

CHAPTER 2

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 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
```

```
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 Connecting via Cloud ID

Cloud ID is an easy way to configure your client to work with your Elastic Cloud deployment. Combine the `cloud_id` with either `http_auth` or `api_key` to authenticate with your Elastic Cloud deployment.

Using `cloud_id` enables TLS verification and HTTP compression by default and sets the port to `443` unless otherwise overwritten via the `port` parameter or the port value encoded within `cloud_id`. Using Cloud ID also disables sniffing.

```
from elasticsearch import Elasticsearch

es = Elasticsearch(
    cloud_id="cluster-1:dXMa5Fx...",
    http_auth=("elastic", "<password>"),
)
```

4.7 API Key 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',
)
```

4.8 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)
```

Compression is enabled by default when connecting to Elastic Cloud via `cloud_id`.

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')

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

Elasticsearch-DSL

For a more high level client library with more limited scope, have a look at elasticsearch-dsl - a more pythonic library sitting on top of `elasticsearch-py`.

elasticsearch-dsl provides a more convenient and idiomatic way to write and manipulate queries by mirroring the terminology and structure of Elasticsearch JSON DSL while exposing the whole range of the DSL from Python either directly using defined classes or a queryset-like expressions.

It also provides an optional persistence layer for working with documents as Python objects in an ORM-like fashion: defining mappings, retrieving and saving documents, wrapping the document data in user-defined classes.

CHAPTER 8

Contents

8.1 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.

8.1.1 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.

Tracking Requests with Opaque ID

You can enrich your requests against Elasticsearch with an identifier string, that allows you to discover this identifier in deprecation logs, to support you with identifying search slow log origin or to help with identifying running tasks.

```
from elasticsearch import Elasticsearch

client = Elasticsearch()

# You can apply X-Opaque-Id in any API request via 'opaque_id':
resp = client.get(index="test", id="1", opaque_id="request-1")
```

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._*'])
```

8.1.2 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
    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(*body*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)

Allows to perform multiple index/update/delete operations in a single request. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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
- **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
- **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(*body=None*, *scroll_id=None*, *params=None*, *headers=None*)
Explicitly clears the search context for a scroll. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/clear-scroll-api.html

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

close()
Closes the Transport and all internal connections

count(*body=None*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)
Returns number of documents matching a query. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*index*, *id*, *body*, *doc_type=None*, *params=None*, *headers=None*)

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/7.9/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(*index*, *id*, *doc_type=None*, *params=None*, *headers=None*)

Removes a document from the index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 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, 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(*index*, *body*, *doc_type=None*, *params=None*, *headers=None*)

Deletes documents matching the provided query. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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)
- **analyzer** – The analyzer to use for the query string
- **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, hidden, 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 Default: 100
- **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. Can be set to *auto*. 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(*task_id*, *params=None*, *headers=None*)

Changes the number of requests per second for a particular Delete By Query operation. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*id*, *params=None*, *headers=None*)
Deletes a script. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/modules-scripting.html

Parameters

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

exists(*index*, *id*, *doc_type=None*, *params=None*, *headers=None*)
Returns information about whether a document exists in an index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index*, *id*, *doc_type=None*, *params=None*, *headers=None*)
Returns information about whether a document source exists in an index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index*, *id*, *body=None*, *doc_type=None*, *params=None*, *headers=None*)

Returns information about why a specific matches (or doesn't match) a query. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*body=None*, *index=None*, *params=None*, *headers=None*)

Returns the information about the capabilities of fields among multiple indices. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/search-field-caps.html

Parameters

- **body** – An index filter specified with the Query DSL
- **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, hidden, 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(*index*, *id*, *doc_type=None*, *params=None*, *headers=None*)
Returns a document. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*id*, *params=None*, *headers=None*)
Returns a script. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/modules-scripting.html

Parameters

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

get_script_context(*params=None*, *headers=None*)
Returns all script contexts. https://www.elastic.co/guide/en/elasticsearch/painless/master/painless-contexts.html

get_script_languages(*params=None*, *headers=None*)
Returns available script types, languages and contexts https://www.elastic.co/guide/en/elasticsearch/reference/7.9/modules-scripting.html

get_source(*index*, *id*, *doc_type=None*, *params=None*, *headers=None*)
Returns the source of a document. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index*, *body*, *doc_type=None*, *id=None*, *params=None*, *headers=None*)
Creates or updates a document in an index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 (*params=None*, *headers=None*)
Returns basic information about the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/index.html

mget (*body*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)
Allows to get multiple documents in one request. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 (*body*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)
Allows to execute several search operations in one request. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 its rewrite method ie. if date filters are mandatory to match but the shard bounds and the query are disjoint.
- **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(*body*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)

Allows to execute several search template operations in one request. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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
- **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(*body=None*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)

Returns multiple termvectors in one request. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*params=None*, *headers=None*)
Returns whether the cluster is running. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/index.html

put_script(*id*, *body*, *context=None*, *params=None*, *headers=None*)
Creates or updates a script. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/modules-scripting.html

Parameters

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

rank_eval(*body*, *index=None*, *params=None*, *headers=None*)
Allows to evaluate the quality of ranked search results over a set of typical search queries https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, none, all Default: open
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **search_type** – Search operation type Valid choices: query_then_fetch, dfs_query_then_fetch

reindex(*body*, *params=None*, *headers=None*)

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/7.9/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 affected 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. Can be set to *auto*. 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(*task_id*, *params=None*, *headers=None*)

Changes the number of requests per second for a particular Reindex operation. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*body=None*, *id=None*, *params=None*, *headers=None*)

Allows to use the Mustache language to pre-render a search definition. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*body=None*, *params=None*, *headers=None*)

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 (*body=None*, *scroll_id=None*, *params=None*, *headers=None*)
Allows to retrieve a large numbers of results from a single search request. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 for scrolled search
- **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

search (*body=None*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)
Returns results matching a query. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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 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 its rewrite method ie. if date filters are mandatory to match but the shard bounds and the query are disjoint.
- **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(*index=None*, *params=None*, *headers=None*)

Returns information about the indices and shards that a search request would be executed against. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*body*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)

Allows to use the Mustache language to pre-render a search definition. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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)
- **ccs_minimize_roundtrips** – Indicates whether network round- trips should be minimized as part of cross-cluster search requests execution Default: true

- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both. Valid choices: open, closed, hidden, 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(*index*, *body=None*, *doc_type=None*, *id=None*, *params=None*, *headers=None*)

Returns information and statistics about terms in the fields of a particular document. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index*, *id*, *body*, *doc_type=None*, *params=None*, *headers=None*)

Updates a document with a script or partial document. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 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
- **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(*index*, *body=None*, *doc_type=None*, *params=None*, *headers=None*)

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/7.9/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, hidden, 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 affected 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 Default: 100
- **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. Can be set to *auto*. 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(*task_id*, *params=None*, *headers=None*)
Changes the number of requests per second for a particular Update By Query operation. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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.

8.1.3 Indices

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

add_block(*index*, *block*, *params=None*, *headers=None*)
Adds a block to an index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/index-modules-blocks.html

Parameters

- **index** – A comma separated list of indices to add a block to
- **block** – The block to add (one of read, write, read_only or metadata)
- **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, hidden, 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

analyze(*body=None*, *index=None*, *params=None*, *headers=None*)
Performs the analysis process on a text and return the tokens breakdown of the text. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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

clear_cache(*index=None*, *params=None*, *headers=None*)
Clears all or specific caches for one or more indices. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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)
- **query** – Clear query caches
- **request** – Clear request cache

clone(*index*, *target*, *body=None*, *params=None*, *headers=None*)
Clones an index https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index*, *params=None*, *headers=None*)
Closes an index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*index*, *body=None*, *params=None*, *headers=None*)
Creates an index with optional settings and mappings. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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.

create_data_stream(*name*, *params=None*, *headers=None*)
Creates a data stream https://www.elastic.co/guide/en/elasticsearch/reference/7.9/data-streams.html

Parameters **name** – The name of the data stream

data_streams_stats(*name=None*, *params=None*, *headers=None*)
Provides statistics on operations happening in a data stream. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/data-streams.html

Parameters **name** – A comma-separated list of data stream names; use *_all* or empty string to perform the operation on all data streams

delete(*index*, *params=None*, *headers=None*)
Deletes an index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*index*, *name*, *params=None*, *headers=None*)
Deletes an alias. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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_data_stream(*name*, *params=None*, *headers=None*)
Deletes a data stream. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/data-streams.html

Parameters **name** – A comma-separated list of data streams to delete; use * to delete all data streams

delete_index_template(*name*, *params=None*, *headers=None*)
Deletes an index template. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/indices-templates.html

Parameters

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

delete_template(*name*, *params=None*, *headers=None*)
Deletes an index template. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/indices-templates.html

Parameters

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

exists(*index*, *params=None*, *headers=None*)
Returns information about whether a particular index exists. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*name*, *index=None*, *params=None*, *headers=None*)
Returns information about whether a particular alias exists. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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_index_template(*name*, *params=None*, *headers=None*)
Returns information about whether a particular index template exists. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/indices-templates.html

Parameters

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

exists_template(*name*, *params=None*, *headers=None*)
Returns information about whether a particular index template exists. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index*, *doc_type*, *params=None*, *headers=None*)
Returns information about whether a particular document type exists. (DEPRECATED) https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*index=None*, *params=None*, *headers=None*)
Performs the flush operation on one or more indices. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*index=None*, *params=None*, *headers=None*)
Performs a synced flush operation on one or more indices. Synced flush is deprecated and will be removed in 8.0. Use flush instead https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index=None*, *params=None*, *headers=None*)
Performs the force merge operation on one or more indices. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*index*, *params=None*, *headers=None*)

Freezes an index. A frozen index has almost no overhead on the cluster (except for maintaining its metadata in memory) and is read-only. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/freeze-index-api.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, hidden, 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(*index*, *params=None*, *headers=None*)

Returns information about one or more indices. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*index=None*, *name=None*, *params=None*, *headers=None*)

Returns an alias. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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_data_stream(*name=None*, *params=None*, *headers=None*)
Returns data streams. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/data-streams.html

Parameters **name** – A comma-separated list of data streams to get; use * to get all data streams

get_field_mapping(*fields*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)
Returns mapping for one or more fields. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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_index_template(*name=None*, *params=None*, *headers=None*)
Returns an index template. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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

get_mapping(*index=None*, *doc_type=None*, *params=None*, *headers=None*)
Returns mappings for one or more indices. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*index=None*, *name=None*, *params=None*, *headers=None*)
Returns settings for one or more indices. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, none, all Default: all
- **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(*name=None*, *params=None*, *headers=None*)
Returns an index template. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index=None*, *params=None*, *headers=None*)
The _upgrade API is no longer useful and will be removed. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, none, all Default: open
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

open(*index*, *params=None*, *headers=None*)
Opens an index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*index*, *name*, *body=None*, *params=None*, *headers=None*)
Creates or updates an alias. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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_index_template(*name*, *body*, *params=None*, *headers=None*)
Creates or updates an index template. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/indices-templates.html

Parameters

- **name** – The name of the template

- **body** – The template definition
- **cause** – User defined reason for creating/updating the index template
- **create** – Whether the index template should only be added if new or can also replace an existing one
- **master_timeout** – Specify timeout for connection to master

put_mapping(*body*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)
Updates the index mappings. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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
- **write_index_only** – When true, applies mappings only to the write index of an alias or data stream

put_settings(*body*, *index=None*, *params=None*, *headers=None*)
Updates the index settings. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*name*, *body*, *params=None*, *headers=None*)
Creates or updates an index template. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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
- **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)

recovery(*index=None*, *params=None*, *headers=None*)
Returns information about ongoing index shard recoveries. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index=None*, *params=None*, *headers=None*)
Performs the refresh operation in one or more indices. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, none, all Default: open
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

reload_search_analyzers(*index*, *params=None*, *headers=None*)
Reloads an index's search analyzers and their resources. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, none, all Default: open
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

resolve_index(*name*, *params=None*, *headers=None*)
Returns information about any matching indices, aliases, and data streams https://www.elastic.co/guide/en/elasticsearch/reference/7.9/indices-resolve-index-api.html

Parameters

- **name** – A comma-separated list of names or wildcard expressions
- **expand_wildcards** – Whether wildcard expressions should get expanded to open or closed indices (default: open) Valid choices: open, closed, hidden, none, all Default: open

rollover(*alias*, *body=None*, *new_index=None*, *params=None*, *headers=None*)
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/7.9/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(*index=None*, *params=None*, *headers=None*)
Provides low-level information about segments in a Lucene index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*index=None*, *params=None*, *headers=None*)
Provides store information for shard copies of indices. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*index*, *target*, *body=None*, *params=None*, *headers=None*)
Allow to shrink an existing index into a new index with fewer primary shards. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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.

simulate_index_template(*name*, *body=None*, *params=None*, *headers=None*)
Simulate matching the given index name against the index templates in the system https://www.elastic.co/guide/en/elasticsearch/reference/7.9/indices-templates.html

Parameters

- **name** – The name of the index (it must be a concrete index name)
- **body** – New index template definition, which will be included in the simulation, as if it already exists in the system
- **cause** – User defined reason for dry-run creating the new template for simulation purposes
- **create** – Whether the index template we optionally defined in the body should only be dry-run added if new or can also replace an existing one
- **master_timeout** – Specify timeout for connection to master

simulate_template(*body=None*, *name=None*, *params=None*, *headers=None*)
Simulate resolving the given template name or body https://www.elastic.co/guide/en/elasticsearch/reference/7.9/indices-templates.html

Parameters

- **body** – New index template definition to be simulated, if no index template name is specified
- **name** – The name of the index template
- **cause** – User defined reason for dry-run creating the new template for simulation purposes
- **create** – Whether the index template we optionally defined in the body should only be dry-run added if new or can also replace an existing one
- **master_timeout** – Specify timeout for connection to master

split(*index*, *target*, *body=None*, *params=None*, *headers=None*)

Allows you to split an existing index into a new index with more primary shards. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index=None*, *metric=None*, *params=None*, *headers=None*)

Provides statistics on operations happening in an index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*index*, *params=None*, *headers=None*)

Unfreezes an index. When a frozen index is unfrozen, the index goes through the normal recovery process and becomes writeable again. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/unfreeze-index-api.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, hidden, 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(*body*, *params=None*, *headers=None*)

Updates index aliases. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/indices-aliases.html

Parameters

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

upgrade(*index=None*, *params=None*, *headers=None*)

The _upgrade API is no longer useful and will be removed. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*body=None*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)
Allows a user to validate a potentially expensive query without executing it. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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.

8.1.4 Ingest

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

delete_pipeline(*id*, *params=None*, *headers=None*)
Deletes a pipeline. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/delete-pipeline-api.html

Parameters

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

- **timeout** – Explicit operation timeout

get_pipeline(*id=None*, *params=None*, *headers=None*)
Returns a pipeline. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*params=None*, *headers=None*)
Returns a list of the built-in patterns. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/grok-processor.html#grok-processor-rest-get

put_pipeline(*id*, *body*, *params=None*, *headers=None*)
Creates or updates a pipeline. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*body*, *id=None*, *params=None*, *headers=None*)
Allows to simulate a pipeline with example documents. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/simulate-pipeline-api.html

Parameters

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

8.1.5 Cluster

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

allocation_explain(*body=None*, *params=None*, *headers=None*)
Provides explanations for shard allocations in the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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)

delete_component_template(*name*, *params=None*, *headers=None*)
Deletes a component template https://www.elastic.co/guide/en/elasticsearch/reference/7.9/indices-component-template.html

Parameters

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

delete_voting_config_exclusions(*params=None*, *headers=None*)
Clears cluster voting config exclusions. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/voting-config-exclusions.html

Parameters **wait_for_removal** – Specifies whether to wait for all excluded nodes to be removed from the cluster before clearing the voting configuration exclusions list. Default: True

exists_component_template(*name*, *params=None*, *headers=None*)
Returns information about whether a particular component template exist https://www.elastic.co/guide/en/elasticsearch/reference/7.9/indices-component-template.html

Parameters

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

get_component_template(*name=None*, *params=None*, *headers=None*)
Returns one or more component templates https://www.elastic.co/guide/en/elasticsearch/reference/7.9/indices-component-template.html

Parameters

- **name** – The comma separated names of the component templates
- **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_settings(*params=None*, *headers=None*)
Returns cluster settings. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index=None*, *params=None*, *headers=None*)
Returns basic information about the health of the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*params=None*, *headers=None*)

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/7.9/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

post_voting_config_exclusions(*params=None*, *headers=None*)

Updates the cluster voting config exclusions by node ids or node names. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/voting-config-exclusions.html

Parameters

- **node_ids** – A comma-separated list of the persistent ids of the nodes to exclude from the voting configuration. If specified, you may not also specify ?node_names.
- **node_names** – A comma-separated list of the names of the nodes to exclude from the voting configuration. If specified, you may not also specify ?node_ids.
- **timeout** – Explicit operation timeout Default: 30s

put_component_template(*name*, *body*, *params=None*, *headers=None*)

Creates or updates a component template https://www.elastic.co/guide/en/elasticsearch/reference/7.9/indices-component-template.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
- **master_timeout** – Specify timeout for connection to master

- **timeout** – Explicit operation timeout

put_settings (*body*, *params=None*, *headers=None*)
Updates the cluster settings. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 (*params=None*, *headers=None*)
Returns the information about configured remote clusters. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/cluster-remote-info.html

reroute (*body=None*, *params=None*, *headers=None*)
Allows to manually change the allocation of individual shards in the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 (*metric=None*, *index=None*, *params=None*, *headers=None*)
Returns a comprehensive information about the state of the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*node_id=None*, *params=None*, *headers=None*)
Returns high-level overview of cluster statistics. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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

8.1.6 Nodes

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

hot_threads(*node_id=None*, *params=None*, *headers=None*)
Returns information about hot threads on each node in the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*node_id=None*, *metric=None*, *params=None*, *headers=None*)
Returns information about nodes in the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*body=None*, *node_id=None*, *params=None*, *headers=None*)
Reloads secure settings. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/secure-settings.html#reloadable-secure-settings

Parameters

- **body** – An object containing the password for the elasticsearch keystore
- **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(*node_id=None*, *metric=None*, *index_metric=None*, *params=None*, *headers=None*)
Returns statistical information about nodes in the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, indexing_pressure
- **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(*node_id=None*, *metric=None*, *params=None*, *headers=None*)
Returns low-level information about REST actions usage on nodes. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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

8.1.7 Cat

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

aliases(*name=None*, *params=None*, *headers=None*)
Shows information about currently configured aliases to indices including filter and routing infos. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/cat-alias.html

Parameters

- **name** – A comma-separated list of alias names to return
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both. Valid choices: open, closed, hidden, none, all Default: all
- **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(*node_id=None*, *params=None*, *headers=None*)
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/7.9/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 (*index=None*, *params=None*, *headers=None*)
Provides quick access to the document count of the entire cluster, or individual indices. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 (*fields=None*, *params=None*, *headers=None*)
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/7.9/cat-fielddata.html

Parameters

- **fields** – A comma-separated list of fields to return in the output
- **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

health (*params=None*, *headers=None*)
Returns a concise representation of the cluster health. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, h, m, s, ms, micros, nanos
- **ts** – Set to false to disable timestamping Default: True
- **v** – Verbose mode. Display column headers

help (*params=None*, *headers=None*)
Returns help for the Cat APIs. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/cat.html

Parameters

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

indices(*index=None*, *params=None*, *headers=None*)
Returns information about indices: number of primaries and replicas, document counts, disk size, ... https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, kb, m, mb, g, gb, t, tb, p, pb
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both. Valid choices: open, closed, hidden, none, all Default: all
- **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, h, m, s, ms, micros, nanos
- **v** – Verbose mode. Display column headers

master(*params=None*, *headers=None*)
Returns information about the master node. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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

ml_data_frame_analytics(*id=None*, *params=None*, *headers=None*)
Gets configuration and usage information about data frame analytics jobs. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/cat-dfanalytics.html

Parameters

- **id** – The ID of the data frame analytics to fetch
- **allow_no_match** – Whether to ignore if a wildcard expression matches no configs. (This includes *_all* string or when no configs have been specified)
- **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
- **time** – The unit in which to display time values Valid choices: d, h, m, s, ms, micros, nanos
- **v** – Verbose mode. Display column headers

ml_datafeeds(*datafeed_id=None*, *params=None*, *headers=None*)

Gets configuration and usage information about datafeeds. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/cat-datafeeds.html

Parameters

- **datafeed_id** – The ID of the datafeeds stats to fetch
- **allow_no_datafeeds** – Whether to ignore if a wildcard expression matches no datafeeds. (This includes *_all* string or when no datafeeds have been specified)
- **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, h, m, s, ms, micros, nanos
- **v** – Verbose mode. Display column headers

ml_jobs(*job_id=None*, *params=None*, *headers=None*)

Gets configuration and usage information about anomaly detection jobs. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/cat-anomaly-detectors.html

Parameters

- **job_id** – The ID of the jobs stats to fetch
- **allow_no_jobs** – Whether to ignore if a wildcard expression matches no jobs. (This includes *_all* string or when no jobs have been specified)
- **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

- **time** – The unit in which to display time values Valid choices: d, h, m, s, ms, micros, nanos
- **v** – Verbose mode. Display column headers

ml_trained_models (*model_id=None*, *params=None*, *headers=None*)
Gets configuration and usage information about inference trained models. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/cat-trained-model.html

Parameters

- **model_id** – The ID of the trained models stats to fetch
- **allow_no_match** – Whether to ignore if a wildcard expression matches no trained models. (This includes *_all* string or when no trained models have been specified) Default: True
- **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
- **from** – skips a number of trained models
- **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
- **size** – specifies a max number of trained models to get Default: 100
- **time** – The unit in which to display time values Valid choices: d, h, m, s, ms, micros, nanos
- **v** – Verbose mode. Display column headers

nodeattrs (*params=None*, *headers=None*)
Returns information about custom node attributes. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 (*params=None*, *headers=None*)
Returns basic statistics about performance of cluster nodes. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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** – Calculate the selected nodes using the local cluster state rather than 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, h, m, s, ms, micros, nanos
- **v** – Verbose mode. Display column headers

pending_tasks (*params=None*, *headers=None*)

Returns a concise representation of the cluster pending tasks. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, h, m, s, ms, micros, nanos
- **v** – Verbose mode. Display column headers

plugins (*params=None*, *headers=None*)

Returns information about installed plugins across nodes node. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 (*index=None*, *params=None*, *headers=None*)

Returns information about index shard recoveries, both on-going completed. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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
- **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, h, m, s, ms, micros, nanos
- **v** – Verbose mode. Display column headers

repositories(*params=None*, *headers=None*)
Returns information about snapshot repositories registered in the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index=None*, *params=None*, *headers=None*)
Provides low-level information about the segments in the shards of an index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index=None*, *params=None*, *headers=None*)
Provides a detailed view of shard allocation on nodes. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, h, m, s, ms, micros, nanos
- **v** – Verbose mode. Display column headers

snapshots(*repository=None*, *params=None*, *headers=None*)
Returns all snapshots in a specific repository. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, h, m, s, ms, micros, nanos
- **v** – Verbose mode. Display column headers

tasks(*params=None*, *headers=None*)
Returns information about the tasks currently executing on one or more nodes in the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, h, m, s, ms, micros, nanos
- **v** – Verbose mode. Display column headers

templates(*name=None*, *params=None*, *headers=None*)

Returns information about existing templates. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*thread_pool_patterns=None*, *params=None*, *headers=None*)

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/7.9/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

transforms(*transform_id=None*, *params=None*, *headers=None*)
Gets configuration and usage information about transforms. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/cat-transforms.html

Parameters

- **transform_id** – The id of the transform for which to get stats. '_all' or '*' implies all transforms
- **allow_no_match** – Whether to ignore if a wildcard expression matches no transforms. (This includes *_all* string or when no transforms have been specified)
- **format** – a short version of the Accept header, e.g. json, yaml
- **from** – skips a number of transform configs, defaults to 0
- **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
- **size** – specifies a max number of transforms to get, defaults to 100
- **time** – The unit in which to display time values Valid choices: d, h, m, s, ms, micros, nanos
- **v** – Verbose mode. Display column headers

8.1.8 Snapshot

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

cleanup_repository(*repository*, *params=None*, *headers=None*)
Removes stale data from repository. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/clean-up-snapshot-repo-api.html

Parameters

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

create(*repository*, *snapshot*, *body=None*, *params=None*, *headers=None*)
Creates a snapshot in a repository. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*repository*, *body*, *params=None*, *headers=None*)
Creates a repository. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*repository*, *snapshot*, *params=None*, *headers=None*)
Deletes a snapshot. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/modules-snapshots.html

Parameters

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

delete_repository(*repository*, *params=None*, *headers=None*)
Deletes a repository. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/modules-snapshots.html

Parameters

- **repository** – Name of the snapshot repository to unregister. Wildcard (*) patterns are supported.
- **master_timeout** – Explicit operation timeout for connection to master node
- **timeout** – Explicit operation timeout

get(*repository*, *snapshot*, *params=None*, *headers=None*)
Returns information about a snapshot. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*repository=None*, *params=None*, *headers=None*)
Returns information about a repository. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*repository*, *snapshot*, *body=None*, *params=None*, *headers=None*)
Restores a snapshot. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*repository=None*, *snapshot=None*, *params=None*, *headers=None*)
Returns information about the status of a snapshot. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*repository*, *params=None*, *headers=None*)
Verifies a repository. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/modules-snapshots.html

Parameters

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

8.1.9 Tasks

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

cancel(*task_id=None*, *params=None*, *headers=None*)
Cancels a task, if it can be cancelled through an API. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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.
- **wait_for_completion** – Should the request block until the cancellation of the task and its descendant tasks is completed. Defaults to false

get (*task_id=None*, *params=None*, *headers=None*)
Returns information about a task. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 (*params=None*, *headers=None*)
Returns a list of tasks. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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)

8.1.10 Dangling Indices

class elasticsearch.client.**DanglingIndicesClient** (*client*)

delete_dangling_index (*index_uuid*, *params=None*, *headers=None*)
Deletes the specified dangling index https://www.elastic.co/guide/en/elasticsearch/reference/7.9/modules-gateway-dangling-indices.html

Parameters

- **index_uuid** – The UUID of the dangling index
- **accept_data_loss** – Must be set to true in order to delete the dangling index
- **master_timeout** – Specify timeout for connection to master
- **timeout** – Explicit operation timeout

import_dangling_index (*index_uuid*, *params=None*, *headers=None*)
Imports the specified dangling index https://www.elastic.co/guide/en/elasticsearch/reference/7.9/modules-gateway-dangling-indices.html

Parameters

- **index_uuid** – The UUID of the dangling index
- **accept_data_loss** – Must be set to true in order to import the dangling index
- **master_timeout** – Specify timeout for connection to master

- **timeout** – Explicit operation timeout

list_dangling_indices(*params=None*, *headers=None*)
Returns all dangling indices. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/modules-gateway-dangling-indices.html

8.2 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.

8.2.1 Info

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

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

info(*params=None*, *headers=None*)
Retrieves information about the installed X-Pack features. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/info-api.html

Parameters

- **accept_enterprise** – If an enterprise license is installed, return the type and mode as 'enterprise' (default: false)
- **categories** – Comma-separated list of info categories. Can be any of: build, license, features

usage(*params=None*, *headers=None*)
Retrieves usage information about the installed X-Pack features. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/usage-api.html

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

8.2.2 Async Search APIs

Async Search API lets you asynchronously execute a search request, monitor its progress, and retrieve partial results as they become available.

class elasticsearch.client.async_search.**AsyncSearchClient**(*client*)

delete(*id*, *params=None*, *headers=None*)
Deletes an async search by ID. If the search is still running, the search request will be cancelled. Otherwise, the saved search results are deleted. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/async-search.html

Parameters **id** – The async search ID

get(*id*, *params=None*, *headers=None*)
Retrieves the results of a previously submitted async search request given its ID. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/async-search.html

Parameters

- **id** – The async search ID
- **keep_alive** – Specify the time interval in which the results (partial or final) for this search will be available
- **typed_keys** – Specify whether aggregation and suggester names should be prefixed by their respective types in the response
- **wait_for_completion_timeout** – Specify the time that the request should block waiting for the final response

submit(*body=None*, *index=None*, *params=None*, *headers=None*)

Executes a search request asynchronously. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/async-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
- **_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 the granularity at which progress results will be made available. Default: 5
- **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, hidden, 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)

- **keep_alive** – Update the time interval in which the results (partial or final) for this search will be available Default: 5d
- **keep_on_completion** – Control whether the response should be stored in the cluster if it completed within the provided [wait_for_completion] time (default: false)
- **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
- **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 true
- **routing** – A comma-separated list of specific routing values
- **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
- **wait_for_completion_timeout** – Specify the time that the request should block waiting for the final response Default: 1s

8.2.3 Autoscaling APIs

Autoscaling API gets the current autoscaling decision based on the configured autoscaling policy.

class elasticsearch.client.autoscaling.**AutoscalingClient**(*client*)

delete_autoscaling_policy(*name*, *params=None*, *headers=None*)
Deletes an autoscaling policy. https://www.elastic.co/guide/en/elasticsearch/reference/current/autoscaling-delete-autoscaling-policy.html

Parameters **name** – the name of the autoscaling policy

get_autoscaling_decision(*params=None*, *headers=None*)
Gets the current autoscaling decision based on the configured autoscaling policy, indicating whether or not autoscaling is needed. https://www.elastic.co/guide/en/elasticsearch/reference/current/autoscaling-get-autoscaling-decision.html

get_autoscaling_policy(*name*, *params=None*, *headers=None*)
Retrieves an autoscaling policy. https://www.elastic.co/guide/en/elasticsearch/reference/current/autoscaling-get-autoscaling-policy.html

Parameters **name** – the name of the autoscaling policy

put_autoscaling_policy(*name*, *body*, *params=None*, *headers=None*)
Creates a new autoscaling policy. https://www.elastic.co/guide/en/elasticsearch/reference/current/autoscaling-put-autoscaling-policy.html

Parameters

- **name** – the name of the autoscaling policy
- **body** – the specification of the autoscaling policy

8.2.4 EQL APIs

EQL API allows querying with Event Query Language (EQL) to search logs and events and match them with shared properties.

class elasticsearch.client.eql.**EqlClient**(*client*)

delete(*id*, *params=None*, *headers=None*)
Deletes an async EQL search by ID. If the search is still running, the search request will be cancelled. Otherwise, the saved search results are deleted. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/eql-search-api.html

Parameters **id** – The async search ID

get(*id*, *params=None*, *headers=None*)
Returns async results from previously executed Event Query Language (EQL) search https://www.elastic.co/guide/en/elasticsearch/reference/7.9/eql-search-api.html

Parameters

- **id** – The async search ID
- **keep_alive** – Update the time interval in which the results (partial or final) for this search will be available Default: 5d
- **wait_for_completion_timeout** – Specify the time that the request should block waiting for the final response

search(*index*, *body*, *params=None*, *headers=None*)
Returns results matching a query expressed in Event Query Language (EQL) https://www.elastic.co/guide/en/elasticsearch/reference/7.9/eql-search-api.html

Parameters

- **index** – The name of the index to scope the operation
- **body** – Eql request body. Use the *query* to limit the query scope.
- **keep_alive** – Update the time interval in which the results (partial or final) for this search will be available Default: 5d
- **keep_on_completion** – Control whether the response should be stored in the cluster if it completed within the provided [wait_for_completion] time (default: false)
- **wait_for_completion_timeout** – Specify the time that the request should block waiting for the final response

8.2.5 Graph Explore APIs

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

class elasticsearch.client.graph.**GraphClient**(*client*)

explore(*index*, *body=None*, *doc_type=None*, *params=None*, *headers=None*)
Explore extracted and summarized information about the documents and terms in an index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/graph-explore-api.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** – Graph Query DSL
- **doc_type** – A comma-separated list of document types to search; leave empty to perform the operation on all types
- **routing** – Specific routing value
- **timeout** – Explicit operation timeout

8.2.6 Licensing APIs

Licensing API can be used to manage your licences.

class elasticsearch.client.license.**LicenseClient**(*client*)

delete(*params=None*, *headers=None*)
Deletes licensing information for the cluster https://www.elastic.co/guide/en/elasticsearch/reference/7.9/delete-license.html

get(*params=None*, *headers=None*)
Retrieves licensing information for the cluster https://www.elastic.co/guide/en/elasticsearch/reference/7.9/get-license.html

Parameters

- **accept_enterprise** – If the active license is an enterprise license, return type as 'enterprise' (default: false)
- **local** – Return local information, do not retrieve the state from master node (default: false)

get_basic_status(*params=None*, *headers=None*)
Retrieves information about the status of the basic license. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/get-basic-status.html

get_trial_status(*params=None*, *headers=None*)
Retrieves information about the status of the trial license. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/get-trial-status.html

post(*body=None*, *params=None*, *headers=None*)
Updates the license for the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/update-license.html

Parameters

- **body** – licenses to be installed
- **acknowledge** – whether the user has acknowledged acknowledge messages (default: false)

post_start_basic(*params=None*, *headers=None*)
Starts an indefinite basic license. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/start-basic.html

Parameters **acknowledge** – whether the user has acknowledged acknowledge messages (default: false)

post_start_trial(*params=None*, *headers=None*)
starts a limited time trial license. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/start-trial.html

Parameters

- **acknowledge** – whether the user has acknowledged acknowledge messages (default: false)
- **doc_type** – The type of trial license to generate (default: "trial")

8.2.7 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.

class elasticsearch.client.ml.**MlClient**(*client*)

close_job(*job_id*, *body=None*, *params=None*, *headers=None*)
Closes one or more anomaly detection jobs. A job can be opened and closed multiple times throughout its lifecycle. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-close-job.html

Parameters

- **job_id** – The name of the job to close
- **body** – The URL params optionally sent in the body
- **allow_no_jobs** – Whether to ignore if a wildcard expression matches no jobs. (This includes *_all* string or when no jobs have been specified)

- **force** – True if the job should be forcefully closed
- **timeout** – Controls the time to wait until a job has closed. Default to 30 minutes

delete_calendar(*calendar_id*, *params=None*, *headers=None*)
Deletes a calendar. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-delete-calendar.html

Parameters **calendar_id** – The ID of the calendar to delete

delete_calendar_event(*calendar_id*, *event_id*, *params=None*, *headers=None*)
Deletes scheduled events from a calendar. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-delete-calendar-event.html

Parameters

- **calendar_id** – The ID of the calendar to modify
- **event_id** – The ID of the event to remove from the calendar

delete_calendar_job(*calendar_id*, *job_id*, *params=None*, *headers=None*)
Deletes anomaly detection jobs from a calendar. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-delete-calendar-job.html

Parameters

- **calendar_id** – The ID of the calendar to modify
- **job_id** – The ID of the job to remove from the calendar

delete_data_frame_analytics(*id*, *params=None*, *headers=None*)
Deletes an existing data frame analytics job. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/delete-dfanalytics.html

Parameters

- **id** – The ID of the data frame analytics to delete
- **force** – True if the job should be forcefully deleted
- **timeout** – Controls the time to wait until a job is deleted. Defaults to 1 minute

delete_datafeed(*datafeed_id*, *params=None*, *headers=None*)
Deletes an existing datafeed. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-delete-datafeed.html

Parameters

- **datafeed_id** – The ID of the datafeed to delete
- **force** – True if the datafeed should be forcefully deleted

delete_expired_data(*body=None*, *job_id=None*, *params=None*, *headers=None*)
Deletes expired and unused machine learning data. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-delete-expired-data.html

Parameters

- **body** – deleting expired data parameters
- **job_id** – The ID of the job(s) to perform expired data hygiene for
- **requests_per_second** – The desired requests per second for the deletion processes.
- **timeout** – How long can the underlying delete processes run until they are canceled

delete_filter(*filter_id*, *params=None*, *headers=None*)
Deletes a filter. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-delete-filter.html

Parameters **filter_id** – The ID of the filter to delete

delete_forecast(*job_id*, *forecast_id=None*, *params=None*, *headers=None*)
Deletes forecasts from a machine learning job. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-delete-forecast.html

Parameters

- **job_id** – The ID of the job from which to delete forecasts
- **forecast_id** – The ID of the forecast to delete, can be comma delimited list. Leaving blank implies *_all*
- **allow_no_forecasts** – Whether to ignore if *_all* matches no forecasts
- **timeout** – Controls the time to wait until the forecast(s) are deleted. Default to 30 seconds

delete_job(*job_id*, *params=None*, *headers=None*)
Deletes an existing anomaly detection job. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-delete-job.html

Parameters

- **job_id** – The ID of the job to delete
- **force** – True if the job should be forcefully deleted
- **wait_for_completion** – Should this request wait until the operation has completed before returning Default: True

delete_model_snapshot(*job_id*, *snapshot_id*, *params=None*, *headers=None*)
Deletes an existing model snapshot. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-delete-snapshot.html

Parameters

- **job_id** – The ID of the job to fetch
- **snapshot_id** – The ID of the snapshot to delete

delete_trained_model(*model_id*, *params=None*, *headers=None*)
Deletes an existing trained inference model that is currently not referenced by an ingest pipeline. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/delete-inference.html

Parameters **model_id** – The ID of the trained model to delete

estimate_model_memory(*body*, *params=None*, *headers=None*)
Estimates the model memory https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-apis.html

Parameters **body** – The analysis config, plus cardinality estimates for fields it references

evaluate_data_frame(*body*, *params=None*, *headers=None*)
Evaluates the data frame analytics for an annotated index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/evaluate-dfanalytics.html

Parameters **body** – The evaluation definition

explain_data_frame_analytics(*body=None*, *id=None*, *params=None*, *headers=None*)
Explains a data frame analytics config. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/explain-dfanalytics.html

Parameters

- **body** – The data frame analytics config to explain

- **id** – The ID of the data frame analytics to explain

find_file_structure(*body*, *params=None*, *headers=None*)
Finds the structure of a text file. The text file must contain data that is suitable to be ingested into Elasticsearch. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-find-file-structure.html

Parameters

- **body** – The contents of the file to be analyzed
- **charset** – Optional parameter to specify the character set of the file
- **column_names** – Optional parameter containing a comma separated list of the column names for a delimited file
- **delimiter** – Optional parameter to specify the delimiter character for a delimited file - must be a single character
- **explain** – Whether to include a commentary on how the structure was derived
- **format** – Optional parameter to specify the high level file format Valid choices: ndjson, xml, delimited, semi_structured_text
- **grok_pattern** – Optional parameter to specify the Grok pattern that should be used to extract fields from messages in a semi- structured text file
- **has_header_row** – Optional parameter to specify whether a delimited file includes the column names in its first row
- **line_merge_size_limit** – Maximum number of characters permitted in a single message when lines are merged to create messages. Default: 10000
- **lines_to_sample** – How many lines of the file should be included in the analysis Default: 1000
- **quote** – Optional parameter to specify the quote character for a delimited file - must be a single character
- **should_trim_fields** – Optional parameter to specify whether the values between delimiters in a delimited file should have whitespace trimmed from them
- **timeout** – Timeout after which the analysis will be aborted Default: 25s
- **timestamp_field** – Optional parameter to specify the timestamp field in the file
- **timestamp_format** – Optional parameter to specify the timestamp format in the file - may be either a Joda or Java time format

flush_job(*job_id*, *body=None*, *params=None*, *headers=None*)
Forces any buffered data to be processed by the job. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-flush-job.html

Parameters

- **job_id** – The name of the job to flush
- **body** – Flush parameters
- **advance_time** – Advances time to the given value generating results and updating the model for the advanced interval
- **calc_interim** – Calculates interim results for the most recent bucket or all buckets within the latency period
- **end** – When used in conjunction with calc_interim, specifies the range of buckets on which to calculate interim results

- **skip_time** – Skips time to the given value without generating results or updating the model for the skipped interval
- **start** – When used in conjunction with calc_interim, specifies the range of buckets on which to calculate interim results

forecast(*job_id*, *params=None*, *headers=None*)

Predicts the future behavior of a time series by using its historical behavior. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-forecast.html

Parameters

- **job_id** – The ID of the job to forecast for
- **duration** – The duration of the forecast
- **expires_in** – The time interval after which the forecast expires. Expired forecasts will be deleted at the first opportunity.
- **max_model_memory** – The max memory able to be used by the forecast. Default is 20mb.

get_buckets(*job_id*, *body=None*, *timestamp=None*, *params=None*, *headers=None*)

Retrieves anomaly detection job results for one or more buckets. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-get-bucket.html

Parameters

- **job_id** – ID of the job to get bucket results from
- **body** – Bucket selection details if not provided in URI
- **timestamp** – The timestamp of the desired single bucket result
- **anomaly_score** – Filter for the most anomalous buckets
- **desc** – Set the sort direction
- **end** – End time filter for buckets
- **exclude_interim** – Exclude interim results
- **expand** – Include anomaly records
- **from** – skips a number of buckets
- **size** – specifies a max number of buckets to get
- **sort** – Sort buckets by a particular field
- **start** – Start time filter for buckets

get_calendar_events(*calendar_id*, *params=None*, *headers=None*)

Retrieves information about the scheduled events in calendars. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-get-calendar-event.html

Parameters

- **calendar_id** – The ID of the calendar containing the events
- **end** – Get events before this time
- **from** – Skips a number of events
- **job_id** – Get events for the job. When this option is used calendar_id must be '_all'
- **size** – Specifies a max number of events to get

- **start** – Get events after this time

get_calendars(*body=None*, *calendar_id=None*, *params=None*, *headers=None*)
Retrieves configuration information for calendars. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-get-calendar.html

Parameters

- **body** – The from and size parameters optionally sent in the body
- **calendar_id** – The ID of the calendar to fetch
- **from** – skips a number of calendars
- **size** – specifies a max number of calendars to get

get_categories(*job_id*, *body=None*, *category_id=None*, *params=None*, *headers=None*)
Retrieves anomaly detection job results for one or more categories. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-get-category.html

Parameters

- **job_id** – The name of the job
- **body** – Category selection details if not provided in URI
- **category_id** – The identifier of the category definition of interest
- **from** – skips a number of categories
- **partition_field_value** – Specifies the partition to retrieve categories for. This is optional, and should never be used for jobs where per-partition categorization is disabled.
- **size** – specifies a max number of categories to get

get_data_frame_analytics(*id=None*, *params=None*, *headers=None*)
Retrieves configuration information for data frame analytics jobs. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/get-dfanalytics.html

Parameters

- **id** – The ID of the data frame analytics to fetch
- **allow_no_match** – Whether to ignore if a wildcard expression matches no data frame analytics. (This includes *_all* string or when no data frame analytics have been specified) Default: True
- **from** – skips a number of analytics
- **size** – specifies a max number of analytics to get Default: 100

get_data_frame_analytics_stats(*id=None*, *params=None*, *headers=None*)
Retrieves usage information for data frame analytics jobs. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/get-dfanalytics-stats.html

Parameters

- **id** – The ID of the data frame analytics stats to fetch
- **allow_no_match** – Whether to ignore if a wildcard expression matches no data frame analytics. (This includes *_all* string or when no data frame analytics have been specified) Default: True
- **from** – skips a number of analytics
- **size** – specifies a max number of analytics to get Default: 100

get_datafeed_stats(*datafeed_id=None*, *params=None*, *headers=None*)
Retrieves usage information for datafeeds. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-get-datafeed-stats.html

Parameters

- **datafeed_id** – The ID of the datafeeds stats to fetch
- **allow_no_datafeeds** – Whether to ignore if a wildcard expression matches no datafeeds. (This includes *_all* string or when no datafeeds have been specified)

get_datafeeds(*datafeed_id=None*, *params=None*, *headers=None*)
Retrieves configuration information for datafeeds. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-get-datafeed.html

Parameters

- **datafeed_id** – The ID of the datafeeds to fetch
- **allow_no_datafeeds** – Whether to ignore if a wildcard expression matches no datafeeds. (This includes *_all* string or when no datafeeds have been specified)

get_filters(*filter_id=None*, *params=None*, *headers=None*)
Retrieves filters. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-get-filter.html

Parameters

- **filter_id** – The ID of the filter to fetch
- **from** – skips a number of filters
- **size** – specifies a max number of filters to get

get_influencers(*job_id*, *body=None*, *params=None*, *headers=None*)
Retrieves anomaly detection job results for one or more influencers. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-get-influencer.html

Parameters

- **job_id** – Identifier for the anomaly detection job
- **body** – Influencer selection criteria
- **desc** – whether the results should be sorted in decending order
- **end** – end timestamp for the requested influencers
- **exclude_interim** – Exclude interim results
- **from** – skips a number of influencers
- **influencer_score** – influencer score threshold for the requested influencers
- **size** – specifies a max number of influencers to get
- **sort** – sort field for the requested influencers
- **start** – start timestamp for the requested influencers

get_job_stats(*job_id=None*, *params=None*, *headers=None*)
Retrieves usage information for anomaly detection jobs. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-get-job-stats.html

Parameters

- **job_id** – The ID of the jobs stats to fetch

- **allow_no_jobs** – Whether to ignore if a wildcard expression matches no jobs. (This includes *_all* string or when no jobs have been specified)

get_jobs (*job_id=None*, *params=None*, *headers=None*)
Retrieves configuration information for anomaly detection jobs. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-get-job.html

Parameters

- **job_id** – The ID of the jobs to fetch
- **allow_no_jobs** – Whether to ignore if a wildcard expression matches no jobs. (This includes *_all* string or when no jobs have been specified)

get_model_snapshots (*job_id*, *body=None*, *snapshot_id=None*, *params=None*, *headers=None*)
Retrieves information about model snapshots. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-get-snapshot.html

Parameters

- **job_id** – The ID of the job to fetch
- **body** – Model snapshot selection criteria
- **snapshot_id** – The ID of the snapshot to fetch
- **desc** – True if the results should be sorted in descending order
- **end** – The filter 'end' query parameter
- **from** – Skips a number of documents
- **size** – The default number of documents returned in queries as a string.
- **sort** – Name of the field to sort on
- **start** – The filter 'start' query parameter

get_overall_buckets (*job_id*, *body=None*, *params=None*, *headers=None*)
Retrieves overall bucket results that summarize the bucket results of multiple anomaly detection jobs. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-get-overall-buckets.html

Parameters

- **job_id** – The job IDs for which to calculate overall bucket results
- **body** – Overall bucket selection details if not provided in URI
- **allow_no_jobs** – Whether to ignore if a wildcard expression matches no jobs. (This includes *_all* string or when no jobs have been specified)
- **bucket_span** – The span of the overall buckets. Defaults to the longest job bucket_span
- **end** – Returns overall buckets with timestamps earlier than this time
- **exclude_interim** – If true overall buckets that include interim buckets will be excluded
- **overall_score** – Returns overall buckets with overall scores higher than this value
- **start** – Returns overall buckets with timestamps after this time
- **top_n** – The number of top job bucket scores to be used in the overall_score calculation

get_records (*job_id*, *body=None*, *params=None*, *headers=None*)
Retrieves anomaly records for an anomaly detection job. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-get-record.html

Parameters

- **job_id** – The ID of the job
- **body** – Record selection criteria
- **desc** – Set the sort direction
- **end** – End time filter for records
- **exclude_interim** – Exclude interim results
- **from** – skips a number of records
- **record_score** – Returns records with anomaly scores greater or equal than this value
- **size** – specifies a max number of records to get
- **sort** – Sort records by a particular field
- **start** – Start time filter for records

get_trained_models(*model_id=None*, *params=None*, *headers=None*)
Retrieves configuration information for a trained inference model. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/get-inference.html

Parameters

- **model_id** – The ID of the trained models to fetch
- **allow_no_match** – Whether to ignore if a wildcard expression matches no trained models. (This includes *_all* string or when no trained models have been specified) Default: True
- **decompress_definition** – Should the model definition be decompressed into valid JSON or returned in a custom compressed format. Defaults to true. Default: True
- **for_export** – Omits fields that are illegal to set on model PUT
- **from** – skips a number of trained models
- **include_model_definition** – Should the full model definition be included in the results. These definitions can be large. So be cautious when including them. Defaults to false.
- **size** – specifies a max number of trained models to get Default: 100
- **tags** – A comma-separated list of tags that the model must have.

get_trained_models_stats(*model_id=None*, *params=None*, *headers=None*)
Retrieves usage information for trained inference models. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/get-inference-stats.html

Parameters

- **model_id** – The ID of the trained models stats to fetch
- **allow_no_match** – Whether to ignore if a wildcard expression matches no trained models. (This includes *_all* string or when no trained models have been specified) Default: True
- **from** – skips a number of trained models
- **size** – specifies a max number of trained models to get Default: 100

info(*params=None*, *headers=None*)
Returns defaults and limits used by machine learning. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/get-ml-info.html

open_job(*job_id*, *params=None*, *headers=None*)
Opens one or more anomaly detection jobs. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-open-job.html

Parameters **job_id** – The ID of the job to open

post_calendar_events(*calendar_id*, *body*, *params=None*, *headers=None*)
Posts scheduled events in a calendar. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-post-calendar-event.html

Parameters

- **calendar_id** – The ID of the calendar to modify
- **body** – A list of events

post_data(*job_id*, *body*, *params=None*, *headers=None*)
Sends data to an anomaly detection job for analysis. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-post-data.html

Parameters

- **job_id** – The name of the job receiving the data
- **body** – The data to process
- **reset_end** – Optional parameter to specify the end of the bucket resetting range
- **reset_start** – Optional parameter to specify the start of the bucket resetting range

preview_datafeed(*datafeed_id*, *params=None*, *headers=None*)
Previews a datafeed. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-preview-datafeed.html

Parameters **datafeed_id** – The ID of the datafeed to preview

put_calendar(*calendar_id*, *body=None*, *params=None*, *headers=None*)
Instantiates a calendar. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-put-calendar.html

Parameters

- **calendar_id** – The ID of the calendar to create
- **body** – The calendar details

put_calendar_job(*calendar_id*, *job_id*, *params=None*, *headers=None*)
Adds an anomaly detection job to a calendar. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-put-calendar-job.html

Parameters

- **calendar_id** – The ID of the calendar to modify
- **job_id** – The ID of the job to add to the calendar

put_data_frame_analytics(*id*, *body*, *params=None*, *headers=None*)
Instantiates a data frame analytics job. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/put-dfanalytics.html

Parameters

- **id** – The ID of the data frame analytics to create

- **body** – The data frame analytics configuration

put_datafeed(*datafeed_id*, *body*, *params=None*, *headers=None*)
Instantiates a datafeed. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-put-datafeed.html

Parameters

- **datafeed_id** – The ID of the datafeed to create
- **body** – The datafeed config
- **allow_no_indices** – Ignore if the source indices expressions resolves to no concrete indices (default: true)
- **expand_wildcards** – Whether source index expressions should get expanded to open or closed indices (default: open) Valid choices: open, closed, hidden, none, all
- **ignore_throttled** – Ignore indices that are marked as throttled (default: true)
- **ignore_unavailable** – Ignore unavailable indexes (default: false)

put_filter(*filter_id*, *body*, *params=None*, *headers=None*)
Instantiates a filter. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-put-filter.html

Parameters

- **filter_id** – The ID of the filter to create
- **body** – The filter details

put_job(*job_id*, *body*, *params=None*, *headers=None*)
Instantiates an anomaly detection job. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-put-job.html

Parameters

- **job_id** – The ID of the job to create
- **body** – The job

put_trained_model(*model_id*, *body*, *params=None*, *headers=None*)
Creates an inference trained model. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/put-inference.html

Parameters

- **model_id** – The ID of the trained models to store
- **body** – The trained model configuration

revert_model_snapshot(*job_id*, *snapshot_id*, *body=None*, *params=None*, *headers=None*)
Reverts to a specific snapshot. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-revert-snapshot.html

Parameters

- **job_id** – The ID of the job to fetch
- **snapshot_id** – The ID of the snapshot to revert to
- **body** – Reversion options
- **delete_intervening_results** – Should we reset the results back to the time of the snapshot?

set_upgrade_mode(*params=None*, *headers=None*)
Sets a cluster wide upgrade_mode setting that prepares machine learning indices for an upgrade. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-set-upgrade-mode.html

Parameters

- **enabled** – Whether to enable upgrade_mode ML setting or not. Defaults to false.
- **timeout** – Controls the time to wait before action times out. Defaults to 30 seconds

start_data_frame_analytics(*id*, *body=None*, *params=None*, *headers=None*)
Starts a data frame analytics job. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/start-dfanalytics.html

Parameters

- **id** – The ID of the data frame analytics to start
- **body** – The start data frame analytics parameters
- **timeout** – Controls the time to wait until the task has started. Defaults to 20 seconds

start_datafeed(*datafeed_id*, *body=None*, *params=None*, *headers=None*)
Starts one or more datafeeds. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-start-datafeed.html

Parameters

- **datafeed_id** – The ID of the datafeed to start
- **body** – The start datafeed parameters
- **end** – The end time when the datafeed should stop. When not set, the datafeed continues in real time
- **start** – The start time from where the datafeed should begin
- **timeout** – Controls the time to wait until a datafeed has started. Default to 20 seconds

stop_data_frame_analytics(*id*, *body=None*, *params=None*, *headers=None*)
Stops one or more data frame analytics jobs. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/stop-dfanalytics.html

Parameters

- **id** – The ID of the data frame analytics to stop
- **body** – The stop data frame analytics parameters
- **allow_no_match** – Whether to ignore if a wildcard expression matches no data frame analytics. (This includes *_all* string or when no data frame analytics have been specified)
- **force** – True if the data frame analytics should be forcefully stopped
- **timeout** – Controls the time to wait until the task has stopped. Defaults to 20 seconds

stop_datafeed(*datafeed_id*, *params=None*, *headers=None*)
Stops one or more datafeeds. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-stop-datafeed.html

Parameters

- **datafeed_id** – The ID of the datafeed to stop
- **allow_no_datafeeds** – Whether to ignore if a wildcard expression matches no datafeeds. (This includes *_all* string or when no datafeeds have been specified)

- **force** – True if the datafeed should be forcefully stopped.
- **timeout** – Controls the time to wait until a datafeed has stopped. Default to 20 seconds

update_data_frame_analytics(*id*, *body*, *params=None*, *headers=None*)
Updates certain properties of a data frame analytics job. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/update-dfanalytics.html

Parameters

- **id** – The ID of the data frame analytics to update
- **body** – The data frame analytics settings to update

update_datafeed(*datafeed_id*, *body*, *params=None*, *headers=None*)
Updates certain properties of a datafeed. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-update-datafeed.html

Parameters

- **datafeed_id** – The ID of the datafeed to update
- **body** – The datafeed update settings
- **allow_no_indices** – Ignore if the source indices expressions resolves to no concrete indices (default: true)
- **expand_wildcards** – Whether source index expressions should get expanded to open or closed indices (default: open) Valid choices: open, closed, hidden, none, all
- **ignore_throttled** – Ignore indices that are marked as throttled (default: true)
- **ignore_unavailable** – Ignore unavailable indexes (default: false)

update_filter(*filter_id*, *body*, *params=None*, *headers=None*)
Updates the description of a filter, adds items, or removes items. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-update-filter.html

Parameters

- **filter_id** – The ID of the filter to update
- **body** – The filter update

update_job(*job_id*, *body*, *params=None*, *headers=None*)
Updates certain properties of an anomaly detection job. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-update-job.html

Parameters

- **job_id** – The ID of the job to create
- **body** – The job update settings

update_model_snapshot(*job_id*, *snapshot_id*, *body*, *params=None*, *headers=None*)
Updates certain properties of a snapshot. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ml-update-snapshot.html

Parameters

- **job_id** – The ID of the job to fetch
- **snapshot_id** – The ID of the snapshot to update
- **body** – The model snapshot properties to update

validate(*body*, *params=None*, *headers=None*)
Validates an anomaly detection job. https://www.elastic.co/guide/en/machine-learning/current/ml-jobs.html

Parameters **body** – The job config

validate_detector(*body*, *params=None*, *headers=None*)
Validates an anomaly detection detector. https://www.elastic.co/guide/en/machine-learning/current/ml-jobs.html

Parameters **body** – The detector

8.2.8 Security APIs

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

class elasticsearch.client.security.**SecurityClient**(*client*)

authenticate(*params=None*, *headers=None*)
Enables authentication as a user and retrieve information about the authenticated user. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-authenticate.html

change_password(*body*, *username=None*, *params=None*, *headers=None*)
Changes the passwords of users in the native realm and built-in users. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-change-password.html

Parameters

- **body** – the new password for the user
- **username** – The username of the user to change the password for
- **refresh** – If *true* (the default) 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* then do nothing with refreshes. Valid choices: true, false, wait_for

clear_cached_privileges(*application*, *params=None*, *headers=None*)
Evicts application privileges from the native application privileges cache. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-clear-privilege-cache.html

Parameters **application** – A comma-separated list of application names

clear_cached_realms(*realms*, *params=None*, *headers=None*)
Evicts users from the user cache. Can completely clear the cache or evict specific users. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-clear-cache.html

Parameters

- **realms** – Comma-separated list of realms to clear
- **usernames** – Comma-separated list of usernames to clear from the cache

clear_cached_roles(*name*, *params=None*, *headers=None*)
Evicts roles from the native role cache. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-clear-role-cache.html

Parameters **name** – Role name

create_api_key(*body*, *params=None*, *headers=None*)
Creates an API key for access without requiring basic authentication. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-create-api-key.html

Parameters

- **body** – The api key request to create an API key
- **refresh** – If *true* (the default) 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* then do nothing with refreshes. Valid choices: true, false, wait_for

delete_privileges(*application*, *name*, *params=None*, *headers=None*)
Removes application privileges. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-delete-privilege.html

Parameters

- **application** – Application name
- **name** – Privilege name
- **refresh** – If *true* (the default) 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* then do nothing with refreshes. Valid choices: true, false, wait_for

delete_role(*name*, *params=None*, *headers=None*)
Removes roles in the native realm. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-delete-role.html

Parameters

- **name** – Role name
- **refresh** – If *true* (the default) 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* then do nothing with refreshes. Valid choices: true, false, wait_for

delete_role_mapping(*name*, *params=None*, *headers=None*)
Removes role mappings. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-delete-role-mapping.html

Parameters

- **name** – Role-mapping name
- **refresh** – If *true* (the default) 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* then do nothing with refreshes. Valid choices: true, false, wait_for

delete_user(*username*, *params=None*, *headers=None*)
Deletes users from the native realm. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-delete-user.html

Parameters

- **username** – username
- **refresh** – If *true* (the default) 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* then do nothing with refreshes. Valid choices: true, false, wait_for

disable_user(*username*, *params=None*, *headers=None*)
Disables users in the native realm. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-disable-user.html

Parameters

- **username** – The username of the user to disable

- **refresh** – If *true* (the default) 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* then do nothing with refreshes. Valid choices: true, false, wait_for

enable_user (*username*, *params=None*, *headers=None*)
Enables users in the native realm. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-enable-user.html

Parameters

- **username** – The username of the user to enable
- **refresh** – If *true* (the default) 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* then do nothing with refreshes. Valid choices: true, false, wait_for

get_api_key (*params=None*, *headers=None*)
Retrieves information for one or more API keys. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-get-api-key.html

Parameters

- **id** – API key id of the API key to be retrieved
- **name** – API key name of the API key to be retrieved
- **owner** – flag to query API keys owned by the currently authenticated user
- **realm_name** – realm name of the user who created this API key to be retrieved
- **username** – user name of the user who created this API key to be retrieved

get_builtin_privileges (*params=None*, *headers=None*)
Retrieves the list of cluster privileges and index privileges that are available in this version of Elasticsearch. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-get-builtin-privileges.html

get_privileges (*application=None*, *name=None*, *params=None*, *headers=None*)
Retrieves application privileges. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-get-privileges.html

Parameters

- **application** – Application name
- **name** – Privilege name

get_role (*name=None*, *params=None*, *headers=None*)
Retrieves roles in the native realm. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-get-role.html

Parameters **name** – Role name

get_role_mapping (*name=None*, *params=None*, *headers=None*)
Retrieves role mappings. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-get-role-mapping.html

Parameters **name** – Role-Mapping name

get_token (*body*, *params=None*, *headers=None*)
Creates a bearer token for access without requiring basic authentication. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-get-token.html

Parameters **body** – The token request to get

get_user (*username=None*, *params=None*, *headers=None*)
Retrieves information about users in the native realm and built-in users. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-get-user.html

Parameters **username** – A comma-separated list of usernames

get_user_privileges (*params=None*, *headers=None*)
Retrieves application privileges. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-get-privileges.html

has_privileges (*body*, *user=None*, *params=None*, *headers=None*)
Determines whether the specified user has a specified list of privileges. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-has-privileges.html

Parameters

- **body** – The privileges to test
- **user** – Username

invalidate_api_key (*body*, *params=None*, *headers=None*)
Invalidates one or more API keys. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-invalidate-api-key.html

Parameters **body** – The api key request to invalidate API key(s)

invalidate_token (*body*, *params=None*, *headers=None*)
Invalidates one or more access tokens or refresh tokens. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-invalidate-token.html

Parameters **body** – The token to invalidate

put_privileges (*body*, *params=None*, *headers=None*)
Adds or updates application privileges. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-put-privileges.html

Parameters

- **body** – The privilege(s) to add
- **refresh** – If *true* (the default) 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* then do nothing with refreshes. Valid choices: true, false, wait_for

put_role (*name*, *body*, *params=None*, *headers=None*)
Adds and updates roles in the native realm. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-put-role.html

Parameters

- **name** – Role name
- **body** – The role to add
- **refresh** – If *true* (the default) 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* then do nothing with refreshes. Valid choices: true, false, wait_for

put_role_mapping (*name*, *body*, *params=None*, *headers=None*)
Creates and updates role mappings. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-put-role-mapping.html

Parameters

- **name** – Role-mapping name

- **body** – The role mapping to add
- **refresh** – If *true* (the default) 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* then do nothing with refreshes. Valid choices: true, false, wait_for

put_user(*username*, *body*, *params=None*, *headers=None*)
Adds and updates users in the native realm. These users are commonly referred to as native users. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/security-api-put-user.html

Parameters

- **username** – The username of the User
- **body** – The user to add
- **refresh** – If *true* (the default) 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* then do nothing with refreshes. Valid choices: true, false, wait_for

8.2.9 Watcher APIs

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

class elasticsearch.client.watcher.**WatcherClient**(*client*)

ack_watch(*watch_id*, *action_id=None*, *params=None*, *headers=None*)
Acknowledges a watch, manually throttling the execution of the watch's actions. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/watcher-api-ack-watch.html

Parameters

- **watch_id** – Watch ID
- **action_id** – A comma-separated list of the action ids to be acked

activate_watch(*watch_id*, *params=None*, *headers=None*)
Activates a currently inactive watch. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/watcher-api-activate-watch.html

Parameters **watch_id** – Watch ID

deactivate_watch(*watch_id*, *params=None*, *headers=None*)
Deactivates a currently active watch. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/watcher-api-deactivate-watch.html

Parameters **watch_id** – Watch ID

delete_watch(*id*, *params=None*, *headers=None*)
Removes a watch from Watcher. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/watcher-api-delete-watch.html

Parameters **id** – Watch ID

execute_watch(*body=None*, *id=None*, *params=None*, *headers=None*)
Forces the execution of a stored watch. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/watcher-api-execute-watch.html

Parameters

- **body** – Execution control
- **id** – Watch ID

- **debug** – indicates whether the watch should execute in debug mode

get_watch(*id*, *params=None*, *headers=None*)
Retrieves a watch by its ID. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/watcher-api-get-watch.html

Parameters **id** – Watch ID

put_watch(*id*, *body=None*, *params=None*, *headers=None*)
Creates a new watch, or updates an existing one. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/watcher-api-put-watch.html

Parameters

- **id** – Watch ID
- **body** – The watch
- **active** – Specify whether the watch is in/active by default
- **if_primary_term** – only update the watch if the last operation that has changed the watch has the specified primary term
- **if_seq_no** – only update the watch if the last operation that has changed the watch has the specified sequence number
- **version** – Explicit version number for concurrency control

start(*params=None*, *headers=None*)
Starts Watcher if it is not already running. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/watcher-api-start.html

stats(*metric=None*, *params=None*, *headers=None*)
Retrieves the current Watcher metrics. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/watcher-api-stats.html

Parameters

- **metric** – Controls what additional stat metrics should be include in the response Valid choices: _all, queued_watches, current_watches, pending_watches
- **emit_stacktraces** – Emits stack traces of currently running watches

stop(*params=None*, *headers=None*)
Stops Watcher if it is running. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/watcher-api-stop.html

8.2.10 Migration APIs

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

class elasticsearch.client.migration.**MigrationClient**(*client*)

deprecations(*index=None*, *params=None*, *headers=None*)
Retrieves information about different cluster, node, and index level settings that use deprecated features that will be removed or changed in the next major version. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/migration-api-deprecation.html

Parameters **index** – Index pattern

8.2.11 Enrich APIs

Enrich API can be used to add data from your existing indices to incoming documents during ingest.

class elasticsearch.client.enrich.**EnrichClient**(*client*)

delete_policy(*name*, *params=None*, *headers=None*)
Deletes an existing enrich policy and its enrich index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/delete-enrich-policy-api.html

Parameters **name** – The name of the enrich policy

execute_policy(*name*, *params=None*, *headers=None*)
Creates the enrich index for an existing enrich policy. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/execute-enrich-policy-api.html

Parameters

- **name** – The name of the enrich policy
- **wait_for_completion** – Should the request should block until the execution is complete. Default: True

get_policy(*name=None*, *params=None*, *headers=None*)
Gets information about an enrich policy. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/get-enrich-policy-api.html

Parameters **name** – A comma-separated list of enrich policy names

put_policy(*name*, *body*, *params=None*, *headers=None*)
Creates a new enrich policy. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/put-enrich-policy-api.html

Parameters

- **name** – The name of the enrich policy
- **body** – The enrich policy to register

stats(*params=None*, *headers=None*)
Gets enrich coordinator statistics and information about enrich policies that are currently executing. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/enrich-stats-api.html

8.2.12 SQL APIs

The SQL REST API accepts SQL in a JSON document, executes it, and returns the results.

class elasticsearch.client.sql.**SqlClient**(*client*)

clear_cursor(*body*, *params=None*, *headers=None*)
Clears the SQL cursor https://www.elastic.co/guide/en/elasticsearch/reference/7.9/sql-pagination.html

Parameters **body** – Specify the cursor value in the *cursor* element to clean the cursor.

query(*body*, *params=None*, *headers=None*)
Executes a SQL request https://www.elastic.co/guide/en/elasticsearch/reference/7.9/sql-rest-overview.html

Parameters

- **body** – Use the *query* element to start a query. Use the *cursor* element to continue a query.

- **format** – a short version of the Accept header, e.g. json, yaml

translate(*body*, *params=None*, *headers=None*)
Translates SQL into Elasticsearch queries https://www.elastic.co/guide/en/elasticsearch/reference/7.9/sql-translate.html

Parameters **body** – Specify the query in the *query* element.

8.2.13 Cross-Cluster Replication APIs

Cross-Cluster Replication API used to perform cross-cluster replication operations.

class elasticsearch.client.ccr.**CcrClient**(*client*)

delete_auto_follow_pattern(*name*, *params=None*, *headers=None*)
Deletes auto-follow patterns. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ccr-delete-auto-follow-pattern.html

Parameters **name** – The name of the auto follow pattern.

follow(*index*, *body*, *params=None*, *headers=None*)
Creates a new follower index configured to follow the referenced leader index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ccr-put-follow.html

Parameters

- **index** – The name of the follower index
- **body** – The name of the leader index and other optional ccr related parameters
- **wait_for_active_shards** – Sets the number of shard copies that must be active before returning. Defaults to 0. 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) Default: 0

follow_info(*index*, *params=None*, *headers=None*)
Retrieves information about all follower indices, including parameters and status for each follower index https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ccr-get-follow-info.html

Parameters **index** – A comma-separated list of index patterns; use *_all* to perform the operation on all indices

follow_stats(*index*, *params=None*, *headers=None*)
Retrieves follower stats. return shard-level stats about the following tasks associated with each shard for the specified indices. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ccr-get-follow-stats.html

Parameters **index** – A comma-separated list of index patterns; use *_all* to perform the operation on all indices

forget_follower(*index*, *body*, *params=None*, *headers=None*)
Removes the follower retention leases from the leader. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ccr-post-forget-follower.html

Parameters

- **index** – the name of the leader index for which specified follower retention leases should be removed
- **body** – the name and UUID of the follower index, the name of the cluster containing the follower index, and the alias from the perspective of that cluster for the remote cluster containing the leader index

get_auto_follow_pattern(*name=None*, *params=None*, *headers=None*)
Gets configured auto-follow patterns. Returns the specified auto-follow pattern collection. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ccr-get-auto-follow-pattern.html

Parameters **name** – The name of the auto follow pattern.

pause_auto_follow_pattern(*name*, *params=None*, *headers=None*)
Pauses an auto-follow pattern https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ccr-pause-auto-follow-pattern.html

Parameters **name** – The name of the auto follow pattern that should pause discovering new indices to follow.

pause_follow(*index*, *params=None*, *headers=None*)
Pauses a follower index. The follower index will not fetch any additional operations from the leader index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ccr-post-pause-follow.html

Parameters **index** – The name of the follower index that should pause following its leader index.

put_auto_follow_pattern(*name*, *body*, *params=None*, *headers=None*)
Creates a new named collection of auto-follow patterns against a specified remote cluster. Newly created indices on the remote cluster matching any of the specified patterns will be automatically configured as follower indices. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ccr-put-auto-follow-pattern.html

Parameters

- **name** – The name of the auto follow pattern.
- **body** – The specification of the auto follow pattern

resume_auto_follow_pattern(*name*, *params=None*, *headers=None*)
Resumes an auto-follow pattern that has been paused https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ccr-resume-auto-follow-pattern.html

Parameters **name** – The name of the auto follow pattern to resume discovering new indices to follow.

resume_follow(*index*, *body=None*, *params=None*, *headers=None*)
Resumes a follower index that has been paused https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ccr-post-resume-follow.html

Parameters

- **index** – The name of the follow index to resume following.
- **body** – The name of the leader index and other optional ccr related parameters

stats(*params=None*, *headers=None*)
Gets all stats related to cross-cluster replication. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ccr-get-stats.html

unfollow(*index*, *params=None*, *headers=None*)
Stops the following task associated with a follower index and removes index metadata and settings associated with cross-cluster replication. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ccr-post-unfollow.html

Parameters **index** – The name of the follower index that should be turned into a regular index.

8.2.14 Monitoring APIs

Monitoring API used to collect data from the Elasticsearch nodes, Logstash nodes, Kibana instances, and Beats in your cluster.

class elasticsearch.client.monitoring.**MonitoringClient**(*client*)

bulk(*body*, *doc_type=None*, *params=None*, *headers=None*)
Used by the monitoring features to send monitoring data. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/monitor-elasticsearch-cluster.html

Parameters

- **body** – The operation definition and data (action-data pairs), separated by newlines
- **doc_type** – Default document type for items which don't provide one
- **interval** – Collection interval (e.g., '10s' or '10000ms') of the payload
- **system_api_version** – API Version of the monitored system
- **system_id** – Identifier of the monitored system

8.2.15 Rollup APIs

Rollup API enables searching through rolled-up data using the standard query DSL.

class elasticsearch.client.rollup.**RollupClient**(*client*)

delete_job(*id*, *params=None*, *headers=None*)
Deletes an existing rollup job. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/rollup-delete-job.html

Parameters **id** – The ID of the job to delete

get_jobs(*id=None*, *params=None*, *headers=None*)
Retrieves the configuration, stats, and status of rollup jobs. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/rollup-get-job.html

Parameters **id** – The ID of the job(s) to fetch. Accepts glob patterns, or left blank for all jobs

get_rollup_caps(*id=None*, *params=None*, *headers=None*)
Returns the capabilities of any rollup jobs that have been configured for a specific index or index pattern. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/rollup-get-rollup-caps.html

Parameters **id** – The ID of the index to check rollup capabilities on, or left blank for all jobs

get_rollup_index_caps(*index*, *params=None*, *headers=None*)
Returns the rollup capabilities of all jobs inside of a rollup index (e.g. the index where rollup data is stored). https://www.elastic.co/guide/en/elasticsearch/reference/7.9/rollup-get-rollup-index-caps.html

Parameters **index** – The rollup index or index pattern to obtain rollup capabilities from.

put_job(*id*, *body*, *params=None*, *headers=None*)
Creates a rollup job. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/rollup-put-job.html

Parameters

- **id** – The ID of the job to create
- **body** – The job configuration

rollup_search(*index*, *body*, *doc_type=None*, *params=None*, *headers=None*)
Enables searching rolled-up data using the standard query DSL. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/rollup-search.html

Parameters

- **index** – The indices or index-pattern(s) (containing rollup or regular data) that should be searched
- **body** – The search request body
- **doc_type** – The doc type inside the index
- **rest_total_hits_as_int** – Indicates whether hits.total should be rendered as an integer or an object in the rest search response
- **typed_keys** – Specify whether aggregation and suggester names should be prefixed by their respective types in the response

start_job(*id*, *params=None*, *headers=None*)
Starts an existing, stopped rollup job. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/rollup-start-job.html

Parameters **id** – The ID of the job to start

stop_job(*id*, *params=None*, *headers=None*)
Stops an existing, started rollup job. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/rollup-stop-job.html

Parameters

- **id** – The ID of the job to stop
- **timeout** – Block for (at maximum) the specified duration while waiting for the job to stop. Defaults to 30s.
- **wait_for_completion** – True if the API should block until the job has fully stopped, false if should be executed async. Defaults to false.

8.2.16 Snapshot Lifecycle Management APIs

Snapshot Lifecycle Management API can be used to set up policies to automatically take snapshots and control how long they are retained.

class elasticsearch.client.slm.**SlmClient**(*client*)

delete_lifecycle(*policy_id*, *params=None*, *headers=None*)
Deletes an existing snapshot lifecycle policy. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/slm-api-delete-policy.html

Parameters **policy_id** – The id of the snapshot lifecycle policy to remove

execute_lifecycle(*policy_id*, *params=None*, *headers=None*)
Immediately creates a snapshot according to the lifecycle policy, without waiting for the scheduled time. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/slm-api-execute-lifecycle.html

Parameters **policy_id** – The id of the snapshot lifecycle policy to be executed

execute_retention(*params=None*, *headers=None*)
Deletes any snapshots that are expired according to the policy's retention rules. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/slm-api-execute-retention.html

get_lifecycle(*policy_id=None*, *params=None*, *headers=None*)
Retrieves one or more snapshot lifecycle policy definitions and information about the latest snapshot attempts. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/slm-api-get-policy.html

Parameters **policy_id** – Comma-separated list of snapshot lifecycle policies to retrieve

get_stats(*params=None*, *headers=None*)
Returns global and policy-level statistics about actions taken by snapshot lifecycle management. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/slm-api-get-stats.html

get_status(*params=None*, *headers=None*)
Retrieves the status of snapshot lifecycle management (SLM). https://www.elastic.co/guide/en/elasticsearch/reference/7.9/slm-api-get-status.html

put_lifecycle(*policy_id*, *body=None*, *params=None*, *headers=None*)
Creates or updates a snapshot lifecycle policy. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/slm-api-put-policy.html

Parameters

- **policy_id** – The id of the snapshot lifecycle policy
- **body** – The snapshot lifecycle policy definition to register

start(*params=None*, *headers=None*)
Turns on snapshot lifecycle management (SLM). https://www.elastic.co/guide/en/elasticsearch/reference/7.9/slm-api-start.html

stop(*params=None*, *headers=None*)
Turns off snapshot lifecycle management (SLM). https://www.elastic.co/guide/en/elasticsearch/reference/7.9/slm-api-stop.html

8.2.17 Searchable Snapshots APIs

Searchable Snapshots API used to perform searchable snapshots operations.

class elasticsearch.client.searchable_snapshots.**SearchableSnapshotsClient**(*client*)

clear_cache(*index=None*, *params=None*, *headers=None*)
Clear the cache of searchable snapshots. https://www.elastic.co/guide/en/elasticsearch/reference/current/searchable-snapshots-api-clear-cache.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
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

mount(*repository*, *snapshot*, *body*, *params=None*, *headers=None*)
Mount a snapshot as a searchable index. https://www.elastic.co/guide/en/elasticsearch/reference/current/searchable-snapshots-api-mount-snapshot.html

Parameters

- **repository** – The name of the repository containing the snapshot of the index to mount

- **snapshot** – The name of the snapshot of the index to mount
- **body** – The restore configuration for mounting the snapshot as searchable
- **master_timeout** – Explicit operation timeout for connection to master node
- **wait_for_completion** – Should this request wait until the operation has completed before returning

repository_stats(*repository*, *params=None*, *headers=None*)
Retrieve usage statistics about a snapshot repository. https://www.elastic.co/guide/en/elasticsearch/reference/current/searchable-snapshots-repository-stats.html

Parameters **repository** – The repository for which to get the stats for

stats(*index=None*, *params=None*, *headers=None*)
Retrieve various statistics about searchable snapshots. https://www.elastic.co/guide/en/elasticsearch/reference/current/searchable-snapshots-api-stats.html

Parameters **index** – A comma-separated list of index names

8.2.18 Index Lifecycle Management APIs

Index Lifecycle Management API used to set up policies to automatically manage the index lifecycle.

class elasticsearch.client.ilm.**IlmClient**(*client*)

delete_lifecycle(*policy*, *params=None*, *headers=None*)
Deletes the specified lifecycle policy definition. A currently used policy cannot be deleted. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ilm-delete-lifecycle.html

Parameters **policy** – The name of the index lifecycle policy

explain_lifecycle(*index*, *params=None*, *headers=None*)
Retrieves information about the index's current lifecycle state, such as the currently executing phase, action, and step. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ilm-explain-lifecycle.html

Parameters

- **index** – The name of the index to explain
- **only_errors** – filters the indices included in the response to ones in an ILM error state, implies only_managed
- **only_managed** – filters the indices included in the response to ones managed by ILM

get_lifecycle(*policy=None*, *params=None*, *headers=None*)
Returns the specified policy definition. Includes the policy version and last modified date. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ilm-get-lifecycle.html

Parameters **policy** – The name of the index lifecycle policy

get_status(*params=None*, *headers=None*)
Retrieves the current index lifecycle management (ILM) status. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ilm-get-status.html

move_to_step(*index*, *body=None*, *params=None*, *headers=None*)
Manually moves an index into the specified step and executes that step. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ilm-move-to-step.html

Parameters

- **index** – The name of the index whose lifecycle step is to change

- **body** – The new lifecycle step to move to

put_lifecycle(*policy*, *body=None*, *params=None*, *headers=None*)
Creates a lifecycle policy https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ilm-put-lifecycle.html

Parameters

- **policy** – The name of the index lifecycle policy
- **body** – The lifecycle policy definition to register

remove_policy(*index*, *params=None*, *headers=None*)
Removes the assigned lifecycle policy and stops managing the specified index https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ilm-remove-policy.html

Parameters **index** – The name of the index to remove policy on

retry(*index*, *params=None*, *headers=None*)
Retries executing the policy for an index that is in the ERROR step. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ilm-retry-policy.html

Parameters **index** – The name of the indices (comma-separated) whose failed lifecycle step is to be retry

start(*params=None*, *headers=None*)
Start the index lifecycle management (ILM) plugin. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ilm-start.html

stop(*params=None*, *headers=None*)
Halts all lifecycle management operations and stops the index lifecycle management (ILM) plugin https://www.elastic.co/guide/en/elasticsearch/reference/7.9/ilm-stop.html

8.2.19 Transform APIs

Transform API manages transformation operations from grabbing data from source indices, transforms it, and saves it to a destination index.

class elasticsearch.client.transform.**TransformClient**(*client*)

delete_transform(*transform_id*, *params=None*, *headers=None*)
Deletes an existing transform. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/delete-transform.html

Parameters

- **transform_id** – The id of the transform to delete
- **force** – When *true*, the transform is deleted regardless of its current state. The default value is *false*, meaning that the transform must be *stopped* before it can be deleted.

get_transform(*transform_id=None*, *params=None*, *headers=None*)
Retrieves configuration information for transforms. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/get-transform.html

Parameters

- **transform_id** – The id or comma delimited list of id expressions of the transforms to get, '_all' or '*' implies get all transforms
- **allow_no_match** – Whether to ignore if a wildcard expression matches no transforms. (This includes *_all* string or when no transforms have been specified)

- **from** – skips a number of transform configs, defaults to 0
- **size** – specifies a max number of transforms to get, defaults to 100

get_transform_stats(*transform_id*, *params=None*, *headers=None*)
Retrieves usage information for transforms. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/get-transform-stats.html

Parameters

- **transform_id** – The id of the transform for which to get stats. '_all' or '*' implies all transforms
- **allow_no_match** – Whether to ignore if a wildcard expression matches no transforms. (This includes *_all* string or when no transforms have been specified)
- **from** – skips a number of transform stats, defaults to 0
- **size** – specifies a max number of transform stats to get, defaults to 100

preview_transform(*body*, *params=None*, *headers=None*)
Previews a transform. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/preview-transform.html

Parameters **body** – The definition for the transform to preview

put_transform(*transform_id*, *body*, *params=None*, *headers=None*)
Instantiates a transform. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/put-transform.html

Parameters

- **transform_id** – The id of the new transform.
- **body** – The transform definition
- **defer_validation** – If validations should be deferred until transform starts, defaults to false.

start_transform(*transform_id*, *params=None*, *headers=None*)
Starts one or more transforms. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/start-transform.html

Parameters

- **transform_id** – The id of the transform to start
- **timeout** – Controls the time to wait for the transform to start

stop_transform(*transform_id*, *params=None*, *headers=None*)
Stops one or more transforms. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/stop-transform.html

Parameters

- **transform_id** – The id of the transform to stop
- **allow_no_match** – Whether to ignore if a wildcard expression matches no transforms. (This includes *_all* string or when no transforms have been specified)
- **force** – Whether to force stop a failed transform or not. Default to false
- **timeout** – Controls the time to wait until the transform has stopped. Default to 30 seconds
- **wait_for_checkpoint** – Whether to wait for the transform to reach a checkpoint before stopping. Default to false

- **wait_for_completion** – Whether to wait for the transform to fully stop before returning or not. Default to false

update_transform(*transform_id*, *body*, *params=None*, *headers=None*)
Updates certain properties of a transform. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/update-transform.html

Parameters

- **transform_id** – The id of the transform.
- **body** – The update transform definition
- **defer_validation** – If validations should be deferred until transform starts, defaults to false.

8.2.20 Deprecation APIs

Deprecation API used to retrieve information about different cluster, node, and index level settings that use deprecated features that will be removed or changed in the next major version.

class elasticsearch.client.deprecation.**DeprecationClient**(*client*)

info(*index=None*, *params=None*, *headers=None*)
https://www.elastic.co/guide/en/migration/7.x/migration-api-deprecation.html

Parameters **index** – Index pattern

8.3 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.

8.4 Using Asyncio with Elasticsearch

Starting in `elasticsearch-py` v7.8.0 for Python 3.6+ the `elasticsearch` package supports async/await with Asyncio. Install the package with the `async` extra to install the aiohttp HTTP client and other dependencies required for async support:

```
$ python -m pip install elasticsearch[async]>=7.8.0
```

The same version specifiers for following the Elastic Stack apply to the `async` extra:

```
# Elasticsearch 7.x
$ python -m pip install elasticsearch[async]>=7,<8
```

Note: Async functionality is a new feature of this library in v7.8.0 so please open an issue if you find an issue or have a question about async support.

8.4.1 Getting Started with Async

After installation all async API endpoints are available via *AsyncElasticsearch* and are used in the same way as other APIs, just with an extra `await`:

```
import asyncio
from elasticsearch import AsyncElasticsearch

es = AsyncElasticsearch()

async def main():
    resp = await es.search(
        index="documents",
        body={"query": {"match_all": {}}}
        size=20,
    )
```

```
    print(resp)

loop = asyncio.get_event_loop()
loop.run_until_complete(main())
```

All APIs that are available under the sync client are also available under the async client.

8.4.2 Async Helpers

Async variants of all helpers are available in `elasticsearch.helpers` and are all prefixed with `async_*`. You'll notice that these APIs are identical to the ones in the sync *Helpers* documentation.

All async helpers that accept an iterator or generator also accept async iterators and async generators.

Bulk and Streaming Bulk

```
import asyncio
from elasticsearch import AsyncElasticsearch
from elasticsearch.helpers import async_bulk

es = AsyncElasticsearch()

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

async def main():
    await async_bulk(es, gendata())

loop = asyncio.get_event_loop()
loop.run_until_complete(main())
```

```
import asyncio
from elasticsearch import AsyncElasticsearch
from elasticsearch.helpers import async_bulk

es = AsyncElasticsearch()

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

async def main():
    async for ok, result in async_streaming_bulk(es, gendata()):
        action, result = result.popitem()
```

```
        if not ok:
            print("failed to %s document %s" % ())

loop = asyncio.get_event_loop()
loop.run_until_complete(main())
```

Scan

```
import asyncio
from elasticsearch import AsyncElasticsearch
from elasticsearch.helpers import async_scan

es = AsyncElasticsearch()

async def main():
    async for doc in async_scan(
        client=es,
        query={"query": {"match": {"title": "python"}}},
        index="orders-*"
    ):
        print(doc)

loop = asyncio.get_event_loop()
loop.run_until_complete(main())
```

Reindex

8.4.3 Frequently Asked Questions

NameError / ImportError when importing `AsyncElasticsearch`?

If when trying to use `AsyncElasticsearch` and you're receiving a `NameError` or `ImportError` you should ensure that you're installing via the `elasticsearch[async]` extra given above. The `AsyncElasticsearch` name won't be available unless `aiohttp` and other async dependencies are installed and you're using Python 3.6 or later.

What about the `elasticsearch-async` package?

Previously asyncio was supported separately via the elasticsearch-async package. elasticsearch-async has been deprecated in favor of `elasticsearch` async support. For Elasticsearch 7.x and later you must install `elasticsearch[async]` and use `elasticsearch.AsyncElasticsearch()`.

Receiving 'Unclosed client session / connector' warning?

This warning is created by `aiohttp` when an open HTTP connection is garbage collected. You'll typically run into this when closing your application. To resolve the issue ensure that *`close()`* is called before the *`AsyncElasticsearch`* instance is garbage collected.

For example if using FastAPI that might look like this:

```
from fastapi import FastAPI
from elasticsearch import AsyncElasticsearch

app = FastAPI()
es = AsyncElasticsearch()

# This gets called once the app is shutting down.
@app.on_event("shutdown")
async def app_shutdown():
    await es.close()
```

8.4.4 API Reference

The API of *AsyncElasticsearch* is nearly identical to the API of *Elasticsearch* with the exception that every API call like *search()* is an `async` function and requires an `await` to properly return the response body.

AsyncElasticsearch

Note: To reference Elasticsearch APIs that are namespaced like `.indices.create()` refer to the sync API reference. These APIs are identical between sync and async.

class elasticsearch.**AsyncElasticsearch**(*hosts=None*, *transport_class=<class 'elasticsearch._async.transport.AsyncTransport'>*, ***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
```

```
    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 (*body*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)

Allows to perform multiple index/update/delete operations in a single request. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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
- **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
- **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(*body=None*, *scroll_id=None*, *params=None*, *headers=None*)

Explicitly clears the search context for a scroll. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/clear-scroll-api.html

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

close()

Closes the Transport and all internal connections

count(*body=None*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)

Returns number of documents matching a query. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*index*, *id*, *body*, *doc_type=None*, *params=None*, *headers=None*)

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/7.9/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(*index*, *id*, *doc_type=None*, *params=None*, *headers=None*)

Removes a document from the index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 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, 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(*index*, *body*, *doc_type=None*, *params=None*, *headers=None*)

Deletes documents matching the provided query. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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)
- **analyzer** – The analyzer to use for the query string
- **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, hidden, 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 Default: 100
- **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. Can be set to *auto*. 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(*task_id*, *params=None*, *headers=None*)
Changes the number of requests per second for a particular Delete By Query operation. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*id*, *params=None*, *headers=None*)
Deletes a script. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/modules-scripting.html

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

exists(*index*, *id*, *doc_type=None*, *params=None*, *headers=None*)
Returns information about whether a document exists in an index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index*, *id*, *doc_type=None*, *params=None*, *headers=None*)
Returns information about whether a document source exists in an index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index*, *id*, *body=None*, *doc_type=None*, *params=None*, *headers=None*)

Returns information about why a specific matches (or doesn't match) a query. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*body=None*, *index=None*, *params=None*, *headers=None*)

Returns the information about the capabilities of fields among multiple indices. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/search-field-caps.html

Parameters

- **body** – An index filter specified with the Query DSL
- **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 re-

solves 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, hidden, 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(*index*, *id*, *doc_type=None*, *params=None*, *headers=None*)
Returns a document. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*id*, *params=None*, *headers=None*)
Returns a script. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/modules-scripting.html

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

get_script_context(*params=None*, *headers=None*)
Returns all script contexts. https://www.elastic.co/guide/en/elasticsearch/painless/master/painless-contexts.html

get_script_languages(*params=None*, *headers=None*)
Returns available script types, languages and contexts https://www.elastic.co/guide/en/elasticsearch/reference/7.9/modules-scripting.html

get_source(*index*, *id*, *doc_type=None*, *params=None*, *headers=None*)
Returns the source of a document. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index*, *body*, *doc_type=None*, *id=None*, *params=None*, *headers=None*)

Creates or updates a document in an index. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*params=None*, *headers=None*)

Returns basic information about the cluster. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/index.html

mget(*body*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)

Allows to get multiple documents in one request. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*body*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)

Allows to execute several search operations in one request. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 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 its rewrite method ie. if date filters are mandatory to match but the shard bounds and the query are disjoint.
- **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(*body*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)

Allows to execute several search template operations in one request. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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
- **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(*body=None*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)

Returns multiple termvectors in one request. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, exter-

nal_gte, force

ping(*params=None*, *headers=None*)

Returns whether the cluster is running. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/index.html

put_script(*id*, *body*, *context=None*, *params=None*, *headers=None*)

Creates or updates a script. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/modules-scripting.html

Parameters

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

rank_eval(*body*, *index=None*, *params=None*, *headers=None*)

Allows to evaluate the quality of ranked search results over a set of typical search queries https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, none, all Default: open
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **search_type** – Search operation type Valid choices: query_then_fetch, dfs_query_then_fetch

reindex(*body*, *params=None*, *headers=None*)

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/7.9/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 affected 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. Can be set to *auto*. 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(*task_id*, *params=None*, *headers=None*)

Changes the number of requests per second for a particular Reindex operation. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*body=None*, *id=None*, *params=None*, *headers=None*)

Allows to use the Mustache language to pre-render a search definition. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*body=None*, *params=None*, *headers=None*)

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(*body=None*, *scroll_id=None*, *params=None*, *headers=None*)

Allows to retrieve a large numbers of results from a single search request. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 for scrolled search
- **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

search(*body=None*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)

Returns results matching a query. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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 its rewrite method ie. if date filters are mandatory to match but the shard bounds and the query are disjoint.
- **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(*index=None*, *params=None*, *headers=None*)

Returns information about the indices and shards that a search request would be executed against. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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, hidden, 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(*body*, *index=None*, *doc_type=None*, *params=None*, *headers=None*)

Allows to use the Mustache language to pre-render a search definition. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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)
- **ccs_minimize_roundtrips** – Indicates whether network round- trips should be minimized as part of cross-cluster search requests execution Default: true
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both. Valid choices: open, closed, hidden, none, all Default: open
- **explain** – Specify whether to return detailed information about score computa-

tion 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(*index*, *body=None*, *doc_type=None*, *id=None*, *params=None*, *headers=None*)

Returns information and statistics about terms in the fields of a particular document. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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(*index*, *id*, *body*, *doc_type=None*, *params=None*, *headers=None*)

Updates a document with a script or partial document. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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 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
- **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(*index*, *body=None*, *doc_type=None*, *params=None*, *headers=None*)

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/7.9/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, hidden, 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 affected 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 Default: 100
- **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. Can be set to *auto*. 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(*task_id*, *params=None*, *headers=None*)

Changes the number of requests per second for a particular Update By Query operation. https://www.elastic.co/guide/en/elasticsearch/reference/7.9/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.

AsyncTransport

class elasticsearch.**AsyncTransport**(*hosts*, **args*, *sniff_on_start=False*, ***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.

close()
Explicitly closes connections

create_sniff_task(*initial=False*)
Initiate a sniffing task. Make sure we only have one sniff request running at any given time. If a finished sniffing request is around, collect its result (which can raise its exception).

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

sniff_hosts(*initial=False*)
Either spawns a sniffing_task which does regular sniffing over time or does a single sniffing session and awaits the results.

AIOHttpConnection

class elasticsearch.**AIOHttpConnection**(*host='localhost'*, *port=None*, *http_auth=None*, *use_ssl=False*, *verify_certs=<object object>*, *ssl_show_warn=<object object>*, *ca_certs=None*, *client_cert=None*, *client_key=None*, *ssl_version=None*, *ssl_assert_fingerprint=None*, *maxsize=10*, *headers=None*, *ssl_context=None*, *http_compress=None*, *cloud_id=None*, *api_key=None*, *opaque_id=None*, *loop=None*, ***kwargs*)
Default connection class for `AsyncElasticsearch` using the *aiohttp* library and the http protocol.

Parameters

- **host** – hostname of the node (default: localhost)
- **port** – port to use (integer, default: 9200)
- **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. Convenient way to connect to cloud instances. Other host connection params will be ignored.
- **api_key** – optional API Key authentication as either base64 encoded string or a tuple.
- **opaque_id** – Send this value in the 'X-Opaque-Id' HTTP header For tracing all requests made by this transport.
- **loop** – asyncio Event Loop to use with aiohttp. This is set by default to the currently running loop.

close()
Explicitly closes connection

8.5 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.

8.5.1 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.

DEFAULT_CONNECTION_CLASS

alias of `elasticsearch.connection.http_urllib3.Urllib3HttpConnection`

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

8.5.2 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.

8.5.3 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

8.5.4 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=None*, *http_auth=None*, *use_ssl=False*, *verify_certs=<object object>*, *ssl_show_warn=<object object>*, *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=None*, *cloud_id=None*, *api_key=None*, *opaque_id=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. Convenient way to connect to cloud instances. Other host connection params will be ignored.
- **api_key** – optional API Key authentication as either base64 encoded string or a tuple.
- **opaque_id** – Send this value in the 'X-Opaque-Id' HTTP header For tracing all requests made by this transport.

close()
Explicitly closes connection

8.6 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.

8.6.1 Connection

class elasticsearch.connection.**Connection**(*host='localhost'*, *port=None*, *use_ssl=False*, *url_prefix=''*, *timeout=10*, *headers=None*, *http_compress=None*, *cloud_id=None*, *api_key=None*, *opaque_id=None*, ***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)
- **use_ssl** – use ssl for the connection if *True*
- **url_prefix** – optional url prefix for elasticsearch
- **timeout** – default timeout in seconds (float, default: 10)
- **http_compress** – Use gzip compression
- **cloud_id** – The Cloud ID from ElasticCloud. Convenient way to connect to cloud instances.
- **opaque_id** – Send this value in the 'X-Opaque-Id' HTTP header For tracing all requests made by this transport.

8.6.2 Urllib3HttpConnection

class elasticsearch.connection.**Urllib3HttpConnection**(*host='localhost'*, *port=None*, *http_auth=None*, *use_ssl=False*, *verify_certs=<object object>*, *ssl_show_warn=<object object>*, *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=None*, *cloud_id=None*, *api_key=None*, *opaque_id=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. Convenient way to connect to cloud instances. Other host connection params will be ignored.
- **api_key** – optional API Key authentication as either base64 encoded string or a tuple.
- **opaque_id** – Send this value in the 'X-Opaque-Id' HTTP header For tracing all requests made by this transport.

8.6.3 RequestsHttpConnection

class elasticsearch.connection.**RequestsHttpConnection**(*host='localhost'*, *port=None*, *http_auth=None*, *use_ssl=False*, *verify_certs=True*, *ssl_show_warn=True*, *ca_certs=None*, *client_cert=None*, *client_key=None*, *headers=None*, *http_compress=None*, *cloud_id=None*, *api_key=None*, *opaque_id=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
- **http_compress** – Use gzip compression
- **cloud_id** – The Cloud ID from ElasticCloud. Convenient way to connect to cloud instances. Other host connection params will be ignored.
- **api_key** – optional API Key authentication as either base64 encoded string or a tuple.
- **opaque_id** – Send this value in the 'X-Opaque-Id' HTTP header For tracing all requests made by this transport.

8.7 Helpers

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

8.7.1 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",
            "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.

8.7.2 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"
)
```

8.7.3 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()*

8.8 Changelog

8.8.1 7.9.0 (2020-08-18)

- Added support for ES 7.9 APIs
- Fixed retries to not raise an error when `sniff_on_connection_error=True` and a `TransportError` is raised during the sniff step. Instead the retry will continue or the error that triggered the retry will be raised (See #1279 and #1326)

8.8.2 7.8.0 (2020-06-18)

- Added support for ES 7.8 APIs
- Added support for async/await with asyncio via `AsyncElasticsearch`. See documentation on using Asyncio with Elasticsearch for more information (See #1232, #1235, #1236)
- Added async helpers `async_bulk`, `async_streaming_bulk`, `async_scan`, and `async_reindex` (See #1260)
- Updated `exists_source` API to use non-deprecated Elasticsearch API routes when `doc_type` is not specified to suppress deprecation warnings (See #1272)

8.8.3 7.7.1 (2020-05-26)

- Updated `create`, `update`, `explain`, `get_source`, and `termvectors` APIs to use non-deprecated Elasticsearch API routes when `doc_type` is not specified to suppress deprecation warnings (See #1253)

8.8.4 7.7.0 (2020-05-13)

- Added support for ES 7.7 APIs (See #1182)
- Added `ElasticsearchDeprecationWarning` which is raised when a `Warning` HTTP header is sent by Elasticsearch. (See #1179)
- Added support for serializing `numpy` and `pandas.` data types to `JSONSerializer`. (See #1180)
- Added `certifi` as a dependency so HTTPS connections work automatically.
- Fixed duplicated parameters in some API docstrings (See #1169, thanks to Morten Hauberg!)

8.8.5 7.6.0 (2020-03-19)

- Added support for ES 7.6 APIs
- Added support for X-Opaque-Id to identify long-running tasks
- Added support for HTTP compression to `RequestsHttpConnection`
- Updated default setting of `http_compress` when using `cloud_id` to `True`
- Updated default setting of `sniffing` when using `cloud_id` to `False`
- Updated default port to `443` if `cloud_id` and no other port is defined on the client or within `cloud_id`
- Updated `GET` HTTP requests that contain a body to `POST` where the API allows this to fix proxies rejecting these requests.
- Fix regression of `client.cluster.state()` where the default `metric` should be set to `"_all"` if an index is given (See #1143)
- Fix regression of `client.tasks.get()` without a `task_id` having similar functionality to `client.tasks.list()` This will be removed in `v8.0` of `elasticsearch-py` (See #1157)

8.8.6 7.5.1 (2020-01-19)

- `7.5.0` tag was not released so retagging

8.8.7 7.5.0

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

8.8.8 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

8.8.9 7.0.5 (2019-10-01)

- Fix `verify_certs=False`

8.8.10 7.0.4 (2019-08-22)

- Fix wheel distribution

8.8.11 7.0.3 (2019-08-21)

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

8.8.12 7.0.2 (2019-05-29)

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

8.8.13 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.

8.8.14 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

8.8.15 6.8.1 (2020-03-31)

- Added support for serializing `numpy` and `pandas` data types to `JSONSerializer`. (See #1180)
- Fixed a namespace conflict in `elasticsearch6` wheel distribution for `v6.8.0` (See #1186)

8.8.16 6.8.0 (2020-03-12)

- Added support for HTTP compression to `RequestsHttpConnection`
- Updated cloud_id default port
- Enable HTTP compression and disable sniffing by default when using Cloud ID to connect to ES.
- Updated versioning scheme to match ES major.minor

8.8.17 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

8.8.18 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.

8.8.19 6.1.1 (2018-01-05)

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

8.8.20 6.1.0 (2018-01-05)

- bad release

8.8.21 6.0.0 (2017-11-14)

- compatibility with Elasticsearch 6.0.0

8.8.22 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

8.8.23 5.4.0 (2017-05-18)

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

8.8.24 5.3.0 (2017-03-30)

- Compatibility with elasticsearch 5.3

8.8.25 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.

8.8.26 5.1.0 (2017-01-11)

- Fixed sniffing

8.8.27 5.0.1 (2016-11-02)

- Fixed performance regression in `scan` helper

8.8.28 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

8.8.29 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.

8.8.30 2.3.0 (2016-02-29)

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

8.8.31 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

8.8.32 2.1.0 (2015-10-19)

- move multiprocessing import inside parallel bulk for Google App Engine

8.8.33 2.0.0 (2015-10-14)

- Elasticsearch 2.0 compatibility release

8.8.34 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.

8.8.35 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

8.8.36 1.6.0 (2015-06-10)

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

8.8.37 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

8.8.38 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

8.8.39 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

8.8.40 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

8.8.41 1.1.1 (2014-07-04)

- Bugfix release fixing escaping issues with `request_timeout`.

8.8.42 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

8.8.43 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.

8.8.44 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

8.8.45 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

8.8.46 0.4.2 (2013-10-08)

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

8.8.47 0.4.1 (2013-09-24)

- Initial release.

CHAPTER 9

License

CHAPTER 10

Indices and tables

- genindex
- modindex
- search

Python Module Index

e

Index

D

E

F

G

H

I

L

M

Q

R

S

T

U

V

W

X