## Contents

1 Features 3
2 Basic Example 5
3 Installation 7
4 Dependencies 9
5 Authors and License 11
  5.1 aiues reference 11
  5.2 Glossary 55
6 Indices and tables 57
Python Module Index 59
aioes is a library for working with an Elasticsearch from asyncio framework.
• Easy interface to *ElasticSearch* cluster.

• Modelled after *elasticsearch-py*. Not fully compatible due *yield from* nature of *asyncio* but looks very similar.
import asyncio
from aioes import Elasticsearch

@asyncio.coroutine
def go():
    es = Elasticsearch(['localhost:9200'])
    ret = yield from es.create(index="my-index",
                               doc_type="test-type",
                               id=42,
                               body={"str": "data",
                                     "int": 1})
    assert ret == {'_id': '42',
                   '_index': 'my-index',
                   '_type': 'test-type',
                   '_version': 1,
                   'ok': True})

    answer = yield from es.get(index="my-index",
                               doc_type="test-type",
                               id=42)
    assert answer['_source'] == {'str': 'data', 'int': 1}

loop = asyncio.get_event_loop()
loop.run_until_complete(go())
aioes published on PyPI and can be installed via pip tool:

```
pip3 install aioes
```
CHAPTER 4

Dependencies

• Python 3.3 and asyncio or Python 3.4+
• aiohttp 1.3+
The aioes package is written by Andrew Svetlov. It’s BSD licensed and freely available. Feel free to improve this package and send a pull request to GitHub.

Contents:

**aioes reference**

**Elasticsearch**

class aioes.Elasticsearch
   
   Class for operating on Elasticsearch cluster.

   ping()
      
      A coroutine that returns True if the cluster is up, False otherwise.

      Returns  bool

   info()
      
      A coroutine that returns basic info from the current cluster.

      Returns  resulting JSON

   index(index, doc_type, body, id=None, *, consistency=default, op_type=default, parent=default, refresh=default, replication=default, routing=default, timeout=default, timestamp=default, ttl=default, version=default, version_type=default)
      
      A coroutine that adds or updates a typed JSON document in a specific index, making it searchable.

      Parameters
      
      • **index** – The name of the index
      
      • **doc_type** – The type of the document
      
      • **body** – The document
• **id** – Document ID

• **consistency** – Explicit write consistency setting for the operation

• **op_type** – Explicit operation type (default: index)

• **parent** – ID of the parent document

• **refresh** – Refresh the index after performing the operation

• **replication** – Specific replication type (default: sync)

• **routing** – Specific routing value

• **timeout** – Explicit operation timeout

• **timestamp** – Explicit timestamp for the document

• **ttl** – Expiration time for the document

• **version** – Explicit version number for concurrency control

• **version_type** – Specific version type

**Returns** resulting JSON

See also:


**create** (index, doc_type, body, id=None, *, consistency=default, parent=default, percolate=default, refresh=default, replication=default, routing=default, timeout=default, timestamp=default, ttl=default, version=default, version_type=default)

A coroutine that adds a typed JSON document in a specific index, making it searchable. Behind the scenes this method calls index(..., op_type='create')

**Parameters**

• **index** – The name of the index

• **doc_type** – The type of the document

• **id** – Document ID

• **body** – The document

• **consistency** – Explicit write consistency setting for the operation

• **id** – Specific document ID (when the POST method is used)

• **parent** – ID of the parent document

• **percolate** – Percolator queries to execute while indexing the document

• **refresh** – Refresh the index after performing the operation

• **replication** – Specific replication type (default: sync)

• **routing** – Specific routing value

• **timeout** – Explicit operation timeout

• **timestamp** – Explicit timestamp for the document

• **ttl** – Expiration time for the document

• **version** – Explicit version number for concurrency control

• **version_type** – Specific version type

**Returns** resulting JSON
**exists** *(index, id, doc_type='all', *, parent=default, preference=default, realtime=default, refresh=default, routing=default)*

A coroutine that returns a boolean indicating whether or not given document exists in Elasticsearch.

**Parameters**

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

**Returns** `bool`

**See also:**


**get** *(index, id, doc_type='all', *, _source=default, _source_exclude=default, _source_include=default, fields=default, parent=default, preference=default, realtime=default, refresh=default, routing=default, version=default, version_type=default)*

A coroutine that get a typed JSON document from the index based on its id.

**Parameters**

- **index** – The name of the index
- **id** – The document ID
- **doc_type** – The type of the document (uses _all by default 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_exclude** – A list of fields to exclude from the returned _source field
- **_source_include** – A list of fields to extract and return from the _source field
- **fields** – A comma-separated list of fields to return in the response
- **parent** – The ID of the parent document
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **realtime** – Specify whether to perform the operation in realtime or search mode
- **refresh** – Refresh the shard containing the document before performing the operation
- **routing** – Specific routing value
- **version** – Explicit version number for concurrency control
• **version_type** – Explicit version number for concurrency control

**Returns** resulting JSON

See also:


```
mget (body, index=None, doc_type=None, *, _source=default, _source_exclude=default, _source_include=default, fields=default, parent=default, preference=default, realtime=default, refresh=default, routing=default)
```

A coroutine that get multiple documents based on an index, type (optional) and ids.

**Parameters**

• **body** – Document identifiers; can be either docs (containing full document information) or ids (when index and type is provided in the URL).

• **index** – The name of the index

• **doc_type** – The type of the document

• **_source** – True or false to return the _source field or not, or a list of fields to return

• **_source_exclude** – A list of fields to exclude from the returned _source field

• **_source_include** – A list of fields to extract and return from the _source field

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

• **parent** – The ID of the parent document

• **preference** – Specify the node or shard the operation should be performed on (default: random)

• **realtime** – Specify whether to perform the operation in realtime or search mode

• **refresh** – Refresh the shard containing the document before performing the operation

• **routing** – Specific routing value

**Returns** resulting JSON

See also:


```
get_source (index, id, doc_type='_all', *, _source=default, _source_exclude=default, _source_include=default, parent=default, preference=default, realtime=default, refresh=default, routing=default, version=default, version_type=default)
```

A coroutine that get the source of a document by it’s index, type and id.

**Parameters**

• **index** – The name of the index

• **doc_type** – The type of the document (uses _all by default to fetch the first document matching the ID across all types)

• **id** – The document ID

• **_source** – True or false to return the _source field or not, or a list of fields to return

• **_source_exclude** – A list of fields to exclude from the returned _source field

• **_source_include** – A list of fields to extract and return from the _source field

• **parent** – The ID of the parent document
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **realtime** – Specify whether to perform the operation in realtime or search mode
- **refresh** – Refresh the shard containing the document before performing the operation
- **routing** – Specific routing value
- **version** – Explicit version number for concurrency control
- **version_type** – Explicit version number for concurrency control

**Returns** resulting JSON

See also:


**update** *(index, doc_type, id, body=None, *, consistency=default, fields=default, lang=default, parent=default, refresh=default, replication=default, retry_on_conflict=default, routing=default, script=default, timeout=default, timestamp=default, ttl=default, version=default, version_type=default)*

A coroutine that update a document based on a script or partial data provided.

**Parameters**

- **index** – The name of the index
- **doc_type** – The type of the document
- **id** – Document ID
- **body** – The request definition using either script or partial doc
- **consistency** – Explicit write consistency setting for the operation
- **fields** – A comma-separated list of fields to return in the response
- **lang** – The script language (default: mvel)
- **parent** – ID of the parent document
- **refresh** – Refresh the index after performing the operation
- **replication** – Specific replication type (default: sync)
- **retry_on_conflict** – Specify how many times should the operation be retried when a conflict occurs (default: 0)
- **routing** – Specific routing value
- **script** – The URL-encoded script definition (instead of using request body)
- **timeout** – Explicit operation timeout
- **timestamp** – Explicit timestamp for the document
- **ttl** – Expiration time for the document
- **version** – Explicit version number for concurrency control
- **version_type** – Explicit version number for concurrency control

**Returns** resulting JSON

See also:

search(index=None, doc_type=None, body=None, *, _source=default, _source_exclude=default, _source_include=default, analyze_wildcard=default, analyzer=default, default_operator=default, df=default, explain=default, fields=default, indices_boost=default, lenient=default, allow_no_indices=default, expand_wildcards=default, ignore_unavailable=default, lowercase_expanded_terms=default, from_=default, preference=default, q=default, routing=default, scroll=default, search_type=default, size=default, sort=default, source=default, stats=default, suggest_field=default, suggest_mode=default, suggest_size=default, suggest_text=default, timeout=default, version=default)

A coroutine that execute a search query and get back search hits that match the query.

Parameters

- **index** – A comma-separated list of index names to search; use _all or empty string to perform the operation on all indices
- **doc_type** – A comma-separated list of document types to search; leave empty to perform the operation on all types
- **body** – The search definition using the Query DSL
- **_source** – True or false to return the _source field or not, or a list of fields to return
- **_source_exclude** – A list of fields to exclude from the returned _source field
- **_source_include** – A list of fields to extract and return from the _source field
- **analyze_wildcard** – Specify whether wildcard and prefix queries should be analyzed (default: false)
- **analyzer** – The analyzer to use for the query string
- **default_operator** – The default operator for query string query (AND or OR) (default: OR)
- **df** – The field to use as default where no field prefix is given in the query string
- **explain** – Specify whether to return detailed information about score computation as part of a hit
- **fields** – A comma-separated list of fields to return as part of a hit
- **indices_boost** – Comma-separated list of index boosts
- **lenient** – Specify whether format-based query failures (such as providing text to a numeric field) should be ignored
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes _all string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both, default ‘open’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **lowercase_expanded_terms** – Specify whether query terms should be lowercased
- **from_** – Starting offset (default: 0)
- **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
• **scroll** – Specify how long a consistent view of the index should be maintained for scrolled search

• **search_type** – Search operation type

• **size** – Number of hits to return (default: 10)

• **sort** – A comma-separated list of `<field>:<direction>` pairs

• **source** – The URL-encoded request definition using the Query DSL (instead of using request body)

• **stats** – Specific ‘tag’ of the request for logging and statistical purposes

• **suggest_field** – Specify which field to use for suggestions

• **suggest_mode** – Specify suggest mode (default: missing)

• **suggest_size** – How many suggestions to return in response

• **suggest_text** – The source text for which the suggestions should be returned

• **timeout** – Explicit operation timeout

• **version** – Specify whether to return document version as part of a hit

**Returns** resulting JSON

See also:


**search_shards** *(index=None, doc_type=None, *, allow_no_indices=default, expand_wildcards=default, ignore_unavailable=default, local=default, preference=default, routing=default)*

A coroutine that returns the indices and shards that a search request would be executed against. This can give useful feedback for working out issues or planning optimizations with routing and shard preferences execute a search query and get back search hits that match the query.

**Parameters**

• **index** – The name of the index

• **doc_type** – The type of the document

• **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes `_all` string or when no indices have been specified)

• **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both. (default: “open”)

• **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

**Returns** resulting JSON

See also:

**search_template** *(index=None, doc_type=None, body=None, *, allow_no_indices=default, expand_wildcards=default, ignore_unavailable=default, preference=default, routing=default, scroll=default, search_type=default)*

A coroutine that accepts a body with a query template and a map of key/value pairs to fill in template parameters.

**Parameters**

- **index** – A comma-separated list of index names to search; use `_all` or empty string to perform the operation on all indices
- **doc_type** – A comma-separated list of document types to search; leave empty to perform the operation on all types
- **body** – The search definition template and its params
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes `_all` string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both, default ‘open’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **routing** – A comma-separated list of specific routing values
- **scroll** – Specify how long a consistent view of the index should be maintained for scrolled search
- **search_type** – Search operation type

**Returns** resulting JSON


**explain** *(index, doc_type, id, body=None, *, _source=default, _source_exclude=default, _source_include=default, analyze_wildcard=default, analyzer=default, default_operator=default, df=default, fields=default, lenient=default, lowercase_expanded_terms=default, parent=default, preference=default, q=default, routing=default, source=default)*

A coroutine that computes a score explanation for a query and a specific document. This can give useful feedback whether a document matches or didn’t match a specific query.

**Parameters**

- **index** – The name of the index
- **doc_type** – The type of the document
- **id** – The document ID
- **body** – The query definition using the Query DSL
- **_source** – True or false to return the _source field or not, or a list of fields to return
- **_source_exclude** – A list of fields to exclude from the returned _source field
- **_source_include** – A list of fields to extract and return from the _source field
• **analyze_wildcard** – Specify whether wildcards and prefix queries in the query string query should be analyzed (default: false)

• **analyzer** – The analyzer for the query string query

• **default_operator** – The default operator for query string query (AND or OR), (default: OR)

• **df** – The default field for query string query (default: _all)

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

• **lenient** – Specify whether format-based query failures (such as providing text to a numeric field) should be ignored

• **lowercase_expanded_terms** – Specify whether query terms should be lowercased

• **parent** – The ID of the parent document

• **preference** – Specify the node or shard the operation should be performed on (default: random)

• **q** – Query in the Lucene query string syntax

• **routing** – Specific routing value

• **source** – The URL-encoded query definition (instead of using the request body)

**Returns** resulting JSON

**See also:**

`scroll(scroll_id, *, scroll=default)`
A coroutine that scroll a search request created by specifying the scroll parameter

**Parameters**

• **scroll_id** – The scroll ID

• **scroll** – Specify how long a consistent view of the index should be maintained for scrolled search

**Returns** resulting JSON

**See also:**

`clear_scroll(scroll_id=None, body=None)`
A coroutine that clear the scroll request created by specifying the scroll parameter to search.

**Parameters**

• **scroll_id** – The scroll ID or a list of scroll IDs

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

**Returns** resulting JSON

**See also:**

`delete(index=*, doc_type, id=*, consistency=default, parent=default, refresh=default, replication=default, routing=default, timeout=default, version=default, version_type=default)`
A coroutine that delete a typed JSON document from a specific index based on its id.
Parameters

- **index** – The name of the index
- **doc_type** – The type of the document
- **id** – The document ID
- **consistency** – Specific write consistency setting for the operation
- **parent** – ID of parent document
- **refresh** – Refresh the index after performing the operation
- **replication** – Specific replication type (default: sync)
- **routing** – Specific routing value
- **timeout** – Explicit operation timeout
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type

Returns resulting JSON

See also:


```python
count(index=None, doc_type=None, body=None, *, allow_no_indices=default, expand_wildcards=default, ignore_unavailable=default, min_score=default, preference=default, q=default, routing=default, source=default)
```

A coroutine that execute a query and get the number of matches for that query.

Parameters

- **index** – A comma-separated list of indices to restrict the results
- **doc_type** – A comma-separated list of types to restrict the results
- **body** – A query to restrict the results (optional)
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes _all string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **min_score** – Include only documents with a specific _score value in the result
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **q** – Query in the Lucene query string syntax
- **routing** – Specific routing value
- **source** – The URL-encoded query definition (instead of using the request body)

Returns resulting JSON

See also:

bulk (body, index=None, doc_type=None, *, consistency=default, refresh=default, routing=default, replication=default, timeout=default)

A coroutine that perform many index/delete operations in a single API call.

Parameters

- **body** – The operation definition and data (action-data pairs), as either a newline separated string, or a sequence of dicts to serialize (one per row).
- **index** – Default index for items which don’t provide one
- **doc_type** – Default document type for items which don’t provide one
- **consistency** – Explicit write consistency setting for the operation
- **refresh** – Refresh the index after performing the operation
- **routing** – Specific routing value
- **replication** – Explicitly set the replication type (default: sync)
- **timeout** – Explicit operation timeout

Returns resulting JSON

See also:

msearch (body, index=None, doc_type=None, *, search_type=default)

A coroutine that executes several search requests within the same API.

Parameters

- **body** – The request definitions (metadata-search request definition pairs), as either a newline separated string, or a sequence of dicts to serialize (one per row).
- **index** – A comma-separated list of index names to use as default
- **doc_type** – A comma-separated list of document types to use as default
- **search_type** – Search operation type

Returns resulting JSON

See also:

delete_by_query (index, doc_type=None, body=None, *, allow_no_indices=default, analyzer=default, consistency=default, default_operator=default, df=default, expand_wildcards=default, ignore_unavailable=default, q=default, replication=default, routing=default, source=default, timeout=default)

A coroutine that delete documents from one or more indices and one or more types based on a query.

Parameters

- **index** – A comma-separated list of indices to restrict the operation; use _all to perform the operation on all indices
- **doc_type** – A comma-separated list of types to restrict the operation
- **body** – A query to restrict the operation specified with the Query DSL
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes _all string or when no indices have been specified)
- **analyzer** – The analyzer to use for the query string
suggest(index, body, *, allow_no_indices=default, expand_wildcards=default, ignore_unavailable=default, preference=default, routing=default, source=default)

A coroutine that suggests similar looking terms based on a provided text by using a suggester.

Parameters

- **index** – A comma-separated list of index names to restrict the operation; use _all or empty string to perform the operation on all indices
- **body** – The request definition
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes _all string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **routing** – Specific routing value
- **source** – The URL-encoded request definition (instead of using request body)

Returns resulting JSON

See also:


percolate(index, doc_type, id=None, body=body, *, allow_no_indices=default, expand_wildcards=default, ignore_unavailable=default, percolate_format=default, percolate_index=default, percolate_type=default, preference=default, routing=default, version=default, version_type=default)

A coroutine that allows to register queries against an index, and then send percolate requests which include
a doc, and getting back the queries that match on that doc out of the set of registered queries.

**Parameters**

- **index** – The index of the document being percolated.
- **doc_type** – The type of the document being percolated.
- **id** – Substitute the document in the request body with a document that is known by the specified id. On top of the id, the index and type parameter will be used to retrieve the document from within the cluster.
- **body** – The percolator request definition using the percolate DSL
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes _all string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **percolate_format** – Return an array of matching query IDs instead of objects
- **percolate_index** – The index to percolate the document into. Defaults to index.
- **percolate_type** – The type to percolate document into. Defaults to type.
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **routing** – A comma-separated list of specific routing values
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type

**Returns** resulting JSON

**See also:**


```python
mpercolate(body, index=None, doc_type=None, *, allow_no_indices=default, expand_wildcards=default, ignore_unavailable=default)
```

A coroutine that allows to register queries against an index, and then send percolate requests which include a doc, and getting back the queries that match on that doc out of the set of registered queries.

**Parameters**

- **index** – The index of the document being count percolated to use as default
- **doc_type** – The type of the document being percolated to use as default.
- **body** – The percolate request definitions (header & body pair), separated by newlines
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes _all string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default ‘open’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

**Returns** resulting JSON
count_percolate(index, doc_type, id=None, body=None, *, allow_no_indices=default, expand_wildcards=default, ignore_unavailable=default, percolate_index=default, percolate_type=default, preference=default, routing=default, version=default, version_type=default)

A coroutine that allows to register queries against an index, and then send percolate requests which include a doc, and getting back the queries that match on that doc out of the set of registered queries.

Parameters

- **index** – The index of the document being count percolated.
- **doc_type** – The type of the document being count percolated.
- **id** – Substitute the document in the request body with a document that is known by the specified id. On top of the id, the index and type parameter will be used to retrieve the document from within the cluster.
- **body** – The count percolator request definition using the percolate DSL
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes _all string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open'
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **percolate_index** – The index to count percolate the document into. Defaults to index.
- **percolate_type** – The type to count percolate document into. Defaults to type.
- **preference** – Specify the node or shard the operation should be performed on (default: random)
- **routing** – A comma-separated list of specific routing values
- **version** – Explicit version number for concurrency control
- **version_type** – Specific version type

Returns resulting JSON

mlt(index, doc_type, id, body=None, *, boost_terms=default, include=default, max_doc_freq=default, max_query_terms=default, max_word_length=default, min_doc_freq=default, min_term_freq=default, min_word_length=default, mlt_fields=default, percent_terms_to_match=default, routing=default, search_from=default, search_indices=default, search_query_hint=default, search_scroll=default, search_size=default, search_source=default, search_type=default, search_types=default, stop_words=default)

A coroutine which get documents that are “like” a specified document

Parameters

- **index** – The name of the index
- **doc_type** – The type of the document (use _all to fetch the first document matching the ID across all types)
• **id** – The document ID
• **body** – A specific search request definition
• **boost_terms** – The boost factor
• **include** – Whether to include the queried document from the response
• **max_doc_freq** – The word occurrence frequency as count: words with higher occurrence in the corpus will be ignored
• **max_query_terms** – The maximum query terms to be included in the generated query
• **max_word_length** – The minimum length of the word: longer words will be ignored
• **min_doc_freq** – The word occurrence frequency as count: words with lower occurrence in the corpus will be ignored
• **min_term_freq** – The term frequency as percent: terms with lower occurrence in the source document will be ignored
• **min_word_length** – The minimum length of the word: shorter words will be ignored
• **mlt_fields** – Specific fields to perform the query against
• **percent_terms_to_match** – How many terms have to match in order to consider the document a match (default: 0.3)
• **routing** – Specific routing value
• **search_from** – The offset from which to return results
• **search_indices** – A comma-separated list of indices to perform the query against (default: the index containing the document)
• **search_query_hint** – The search query hint
• **search_scroll** – A scroll search request definition
• **search_size** – The number of documents to return (default: 10)
• **search_source** – A specific search request definition (instead of using the request body)
• **search_type** – Specific search type (eg. *dfs_then_fetch*, *count*, etc)
• **search_types** – A comma-separated list of types to perform the query against (default: the same type as the document)
• **stop_words** – A comma-separated list of stop words to be ignored

**Returns** resulting JSON

See also:


**termvector** *(index, doc_type, id, body=None, *, field_statistics=default, fields=default, offsets=default, parent=default, payloads=default, positions=default, preference=default, routing=default, term_statistics=default)*

A coroutine that returns information and statistics on terms in the fields of a particular document as stored in the index.

**Parameters**

• **index** – The index in which the document resides.
• **doc_type** – The type of the document.
• **id** – The id of the document.

• **body** – Define parameters. See documentation.

• **field_statistics** – Specifies if document count, sum of document frequencies and sum of total term frequencies should be returned. (default True)

• **fields** – A comma-separated list of fields to return.

• **offsets** – Specifies if term offsets should be returned. (default True)

• **parent** – Parent id of documents.

• **payloads** – Specifies if term payloads should be returned. (default True)

• **positions** – Specifies if term positions should be returned. (default True)

• **preference** – Specify the node or shard the operation should be performed on (default: random).

• **routing** – Specific routing value.

• **term_statistics** – Specifies if total term frequency and document frequency should be returned, default False

**Returns** resulting JSON

See also:

**mtermvectors** *(index=None, doc_type=None, body=None, *, field_statistics=default, fields=default, ids=default, offsets=default, parent=default, payloads=default, positions=default, preference=default, routing=default, term_statistics=default)*

A coroutine allows to get multiple termvectors based on an index, type and id.

**Parameters**

• **index** – The index in which the document resides.

• **doc_type** – The type of the document.

• **body** – Define ids, parameters or a list of parameters per document here. You must at least provide a list of document ids. See documentation.

• **field_statistics** – Specifies if document count, sum of document frequencies and sum of total term frequencies should be returned. Applies to all returned documents unless otherwise specified in body “params” or “docs”. (default True)

• **fields** – A comma-separated list of fields to return. Applies to all returned documents unless otherwise specified in body “params” or “docs”.

• **ids** – A comma-separated list of documents ids. You must define ids as parameter or set “ids” or “docs” in the request body

• **offsets** – Specifies if term offsets should be returned. Applies to all returned documents unless otherwise specified in body “params” or “docs”. (default True)

• **parent** – Parent id of documents. Applies to all returned documents unless otherwise specified in body “params” or “docs”.

• **payloads** – Specifies if term payloads should be returned. Applies to all returned documents unless otherwise specified in body “params” or “docs”. (default True)

• **positions** – Specifies if term positions should be returned. Applies to all returned documents unless otherwise specified in body “params” or “docs”. (default True)
• **preference** – Specify the node or shard the operation should be performed on (default: random). Applies to all returned documents unless otherwise specified in body “params” or “docs”.

• **routing** – Specific routing value. Applies to all returned documents unless otherwise specified in body “params” or “docs”.

• **term_statistics** – Specifies if total term frequency and document frequency should be returned. Applies to all returned documents unless otherwise specified in body “params” or “docs”. (default False)

**Returns** resulting JSON

See also:


**benchmark** *(index=None, doc_type=None, body=None, *, verbose=default)*

A coroutine that provides a standard mechanism for submitting queries and measuring their performance relative to one another.

**arg index** A comma-separated list of index names; use _all or empty string to perform the operation on all indices

**arg doc_type** The name of the document type

**arg body** The search definition using the Query DSL

**arg verbose** Specify whether to return verbose statistics about each iteration (default: false)

**returns** resulting JSON

See also:


**put_script** *(lang, id, body)*

A coroutine that creates a script in given language with specified ID.

**Parameters**

• **lang** – Script language

• **id** – Script ID

• **body** – The document

**Returns** resulting JSON

See also:


**get_script** *(lang, id)*

A coroutine that retrieves a script from the API.

**Parameters**

• **lang** – Script language

• **id** – Script ID

**Returns** resulting JSON

See also:


**delete_script** *(lang, id)*

A coroutine that removes a script from the API.

**Parameters**

• **lang** – Script language
- **id** – Script ID

  **Returns** resulting JSON

  See also:


  **put_template** *(id, body)*

  A coroutine that creates a search template with specified ID.

  **Parameters**

  - **id** – Template ID
  - **body** – The document

  **Returns** resulting JSON

  See also:


  **get_template** *(id)*

  A coroutine that retrieves a search template with specified ID.

  **Parameters**

  - **id** – Template ID

  **Returns** resulting JSON

  See also:


  **delete_template** *(id)*

  A coroutine that removes a search template with specified ID.

  **Parameters**

  - **id** – Template ID

  **Returns** resulting JSON

  See also:


### IndicesClient

**Class** `aioes.client.IndicesClient`

Class for operating on Elasticsearch indices.

  **analyze** *(index=None, body=None, *, analyzer=default, char_filters=default, field=default, filters=default, prefer_local=default, text=default, tokenizer=default)*

  A coroutine that make analysis process on a text and return the tokens breakdown of the text.

  **Parameters**

  - **index** – The name of the index
  - **body** – The text on which the analysis should be performed
  - **analyzer** – The name of the analyzer to use
  - **char_filters** – A comma-separated list of character filters to use for the analysis
  - **field** – Use the analyzer configured for this field (instead of passing the analyzer name)
  - **filters** – A comma-separated list of filters to use for the analysis
  - **prefer_local** – With `true`, specify that a local shard should be used if available, with `false`, use a random shard (default: `true`)
  - **text** – The text on which the analysis should be performed (when request body is not used)
  - **tokenizer** – The name of the tokenizer to use for the analysis
create (index, body=None, *, timeout=default, master_timeout=default)
A coroutine that creates an index in Elasticsearch

Parameters

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

Returns resulting JSON

See also:

open (index, *, timeout=default, master_timeout=default, allow_no_indices=default, expand_wildcards=default, ignore_unavailable=default)
A coroutine that open a closed index to make it available for search.

Parameters

• **index** – The name of the index
• **master_timeout** – Specify timeout for connection to master
• **timeout** – Explicit operation timeout
• **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.
• **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

Returns resulting JSON

See also:

close (index, *, allow_no_indices=default, expand_wildcards=default, ignore_unavailable=default, master_timeout=default, timeout=default)
A coroutine that close an index to remove it’s overhead from the cluster. Closed index is blocked for read/write operations.

Parameters

• **index** – The name of the index
• **master_timeout** – Specify timeout for connection to master
• **timeout** – Explicit operation timeout
• **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes _all string or when no indices have been specified)
• **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both, default u'open'

• **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

**Returns** resulting JSON

**See also:**

**delete** *(index, *, master_timeout=default, timeout=default)*
A coroutine that deletes an index.

**Parameters**

• **index** – A comma-separated list of indices to delete; use '_all' or '*' to delete all indices

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

• **timeout** – Explicit operation timeout

**Returns** resulting JSON

**See also:**

**refresh** *(index=None, *, allow_no_indices=default, expand_wildcards=default, ignore_indices=default, ignore_unavailable=default, force=default)*
A coroutine that refresh one or more index, making all operations performed.

**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.

• **ignore_indices** – When performed on multiple indices, allows to ignore missing ones, default u'none'

• **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

• **force** – Force a refresh even if not required

**Returns** resulting JSON

**See also:**

**flush** *(index=None, *, force=default, full=default, allow_no_indices=default, expand_wildcards=default, ignore_indices=default, ignore_unavailable=default)*
A coroutine that flush one or more indices.

**Parameters**

• **index** – A comma-separated list of index names; use _all or empty string to perform the operation on all indices
• **force** – Whether a flush should be forced even if it is not necessarily needed i.e. if no changes will be committed to the index.

• **full** – If set to true a new index writer is created and settings that have been changed related to the index writer will be refreshed.

• **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.

• **ignore_indices** – When performed on multiple indices, allows to ignore missing ones (default: none)

• **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

**Returns** resulting JSON

**See also:**


**exists** *(index, *, allow_no_indices=default, expand_wildcards=default, ignore_unavailable=default, local=default)*

A coroutine that return a boolean indicating whether given index exists.

**Parameters**

• **index** – A list of indices to check

• **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes `_all` string or when no indices have been specified)

• **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default u’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)

**Returns** resulting bool

**See also:**


**exists_type** *(index, doc_type, *, allow_no_indices=default, expand_wildcards=default, ignore_indices=default, ignore_unavailable=default, local=default)*

A coroutine that check if a type/types exists in an index/indices.

**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.
• **ignore_indices** – When performed on multiple indices, allows to ignore *missing* ones
  (default: none)

• **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)

**Returns** resulting bool

**See also:**


**get_settings** *(index=None, name=None, *, expand_wildcards=default, ignore_indices=default, ignore_unavailable=default, flat_settings=default, local=default)*

A coroutine that retrieve settings for one or more (or all) indices.

**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

- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that
  are open, closed or both.

- **ignore_indices** – When performed on multiple indices, allows to ignore *missing*
  ones, default u’none’

- **ignore_unavailable** – Whether specified concrete indices should be ignored when
  unavailable (missing or closed)

- **flat_settings** – Return settings in flat format (default: false)

- **local** – Return local information, do not retrieve the state from master node (default: false)

**Returns** resulting JSON

**See also:**


**put_settings** *(body, index=None, *, allow_no_indices=default, expand_wildcards=default,
  flat_settings=default, ignore_unavailable=default, master_timeout=default)*

A coroutine that change specific index level settings in real time.

**Parameters**

- **body** – The index settings to be updated

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

- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into
  no concrete indices. (This includes _all string or when no indices have been specified)

- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that
  are open, closed or both, default u’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

**Returns**  resulting JSON

**See also:**

`status(index=None, *, allow_no_indices=default, expand_wildcards=default, ignore_indices=default, ignore_unavailable=default, recovery=default, snapshot=default, human=default)`

A **coroutine** that get a comprehensive status information of one or more indices.

**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.
- **ignore_indices** – When performed on multiple indices, allows to ignore missing ones, default u’none’
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **operation_threading** – TODO: ?
- **recovery** – Return information about shard recovery
- **snapshot** – For snapshot status set it to true
- **human** – Whether to return time and byte values in human-readable format.

**Returns**  resulting JSON

**See also:**

`stats(index=None, *, metric=default, completion_fields=default, docs=default, field_data_fields=default, fields=default, groups=default, allow_no_indices=default, expand_wildcards=default, ignore_indices=default, ignore_unavailable=default, human=default, level=default, types=default)`

A **coroutine** that retrieve statistics on different operations happening on an index.

**Parameters**

- **index** – A comma-separated list of index names; use _all or empty string to perform the operation on all indices
- **completion_fields** – A comma-separated list of fields for completion metric (supports wildcards)
- **docs** – the number of docs / deleted docs (docs not yet merged out). Note, affected by refreshing the index
• **fielddata_fields** – A comma-separated list of fields for fielddata metric (supports wildcards)

• **fields** – A comma-separated list of fields for fielddata and completion metric (supports wildcards)

• **groups** – A comma-separated list of search groups for search statistics

• **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.

• **ignore_indices** – When performed on multiple indices, allows to ignore missing ones (default: none)

• **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

• **human** – Whether to return time and byte values in human-readable format.

• **level** – Return stats aggregated at cluster, index or shard level. (“cluster”, “indices” or “shards”, default: “indices”)

• **types** – A comma-separated list of document types for the indexing index metric

**Returns** resulting JSON

**See also:**


```python
segments(index=None, *, allow_no_indices=default, expand_wildcards=default, ignore_indices=default, ignore_unavailable=default, human=default)
```

A coroutine that provide low level segments information that a Lucene index (shard level) is built with.

**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.

• **ignore_indices** – When performed on multiple indices, allows to ignore missing ones, default u’none’

• **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)

• **human** – Whether to return time and byte values in human-readable format (default: false)

**Returns** resulting JSON

**See also:**

**optimize** (index=None, *, flush=default, allow_no_indices=default, expand_wildcards=default, ignore_indices=default, ignore_unavailable=default, max_num_segments=default, only_expunge_deletes=default, operation_threading=default, wait_for_merge=default, force=default)

A coroutine that explicitly optimize one or more indices through an API.

**Parameters**

- **index** – A comma-separated list of index names; use _all or empty string to perform the operation on all indices
- **flush** – Specify whether the index should be flushed after performing the operation (default: true)
- **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.
- **ignore_indices** – When performed on multiple indices, allows to ignore missing ones, default u'none'
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **max_num_segments** – The number of segments the index should be merged into (default: dynamic)
- **only_expunge_deletes** – Specify whether the operation should only expunge deleted documents
- **operation_threading** – TODO: ?
- **wait_for_merge** – Specify whether the request should block until the merge process is finished (default: true)

**Returns** resulting JSON

See also:


**recovery** (index=None, *, active_only=default, detailed=default, human=default)

A coroutine that provides insight into on-going shard recoveries. Recovery status may be reported for specific indices, or cluster-wide.

**Parameters**

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

**Returns** resulting JSON

See also:

clear_cache(index=None, *, field_data=default, fielddata=default, fields=default, filter=default, filter_cache=default, filter_keys=default, id=default, id_cache=default, allow_no_indices=default, expand_wildcards=default, ignore_indices=default, ignore_unavailable=default, recycler=default)

A coroutine that clear either all caches or specific cached associated with one or more indices.

Parameters

- **index** – A comma-separated list of index name to limit the operation
- **field_data** – Clear field data
- **fielddata** – Clear field data
- **fields** – A comma-separated list of fields to clear when using the field_data parameter (default: all)
- **filter** – Clear filter caches
- **filter_cache** – Clear filter caches
- **filter_keys** – A comma-separated list of keys to clear when using the filter_cache parameter (default: all)
- **id** – Clear ID caches for parent/child
- **id_cache** – Clear ID caches for parent/child
- **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.
- **ignore_indices** – When performed on multiple indices, allows to ignore missing ones (default: none)
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
- **index** – A comma-separated list of index name to limit the operation
- **recycler** – Clear the recycler cache

Returns resulting JSON

See also:


validate_query(index=None, doc_type=None, body=None, *, explain=default, allow_no_indices=default, expand_wildcards=default, ignore_indices=default, ignore_unavailable=default, operation_threading=default, q=default, source=default)

A coroutine that validate a potentially expensive query without executing it.

Parameters

- **index** – A comma-separated list of index names to restrict the operation; use _all or empty string to perform the operation on all indices
- **doc_type** – A comma-separated list of document types to restrict the operation; leave empty to perform the operation on all types
- **body** – The query definition
- **explain** – Return detailed information about the error
• **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.
• **ignore_indices** – When performed on multiple indices, allows to ignore missing ones (default: none)
• **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed)
• **operation_threading** – TODO: ?
• **q** – Query in the Lucene query string syntax
• **source** – The URL-encoded query definition (instead of using the request body)

**Returns** resulting JSON

See also:

**put_mapping** *(index, doc_type, body, *, allow_no_indices=default, expand_wildcards=default, ignore_conflicts=default, ignore_unavailable=default, master_timeout=default, timeout=default)*

A coroutine that registers specific mapping definition for a specific type.

**Parameters**

• **index** – A comma-separated list of index names the alias should point to (supports wildcards); use _all or omit to perform the operation on all indices.
• **doc_type** – The name of the document type
• **body** – The mapping definition
• **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes _all string or when no indices have been specified)
• **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both., default u'open'
• **ignore_conflicts** – Specify whether to ignore conflicts while updating the mapping (default: false)
• **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

**Returns** resulting JSON

See also:

**get_mapping** *(index, doc_type=None, *, ignore_unavailable=default, allow_no_indices=default, expand_wildcards=default, local=default)*

A coroutine that retrieves mapping definition of index or index/type.

**Parameters**

• **index** – A comma-separated list of index names; use _all or empty string for all indices
• **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.

• **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)

   **Returns** resulting JSON

   See also:


   **delete_mapping**(index, doc_type, *, master_timeout=default)

   A coroutine that deletes a mapping (type) along with its data

   **Parameters**

   • **index** – A comma-separated list of index names (supports wildcard); use _all for all indices

   • **doc_type** – A comma-separated list of document types to delete (supports wildcards); use _all to delete all document types in the specified indices.

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

   **Returns** resulting JSON

   See also:


   **get_field_mapping**(field, index=None, doc_type=None, *, include_defaults=default, ignore_unavailable=default, allow_no_indices=default, expand_wildcards=default, local=default)

   A coroutine that retrieves mapping definition of a specific field

   **Parameters**

   • **index** – A comma-separated list of index names; use _all or empty string for all indices

   • **doc_type** – A comma-separated list of document types

   • **field** – A comma-separated list of fields to retrieve the mapping for

   • **include_defaults** – A boolean indicating whether to return default values

   • **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.

   • **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)

   **Returns** resulting JSON
See also:

**put_alias** *(name, index=None, body=None, *, timeout=default, master_timeout=default)*

A coroutine that creates an alias for a specific index/indices

**Parameters**

- **index** – A comma-separated list of index names the alias should point to (supports wildcards); use `_all` or omit 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

**Returns** resulting JSON

See also:

**exists_alias** *(name, index=None, *, allow_no_indices=default, expand_wildcards=default, ignore_indices=default, ignoreUnavailable=default, local=default)*

A coroutine that returns a boolean indicating whether given alias exists

**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.
- **ignore_indices** – When performed on multiple indices, allows to ignore missing ones (default: none)
- **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)

**Returns** resulting boolean

See also:

**get_alias** *(index=None, name=None, *, allow_no_indices=default, expand_wildcards=default, ignore_indices=default, ignore_unavailable=default, local=default)*

A coroutine that retrieves a specified alias

**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.

- **ignore_indices** – When performed on multiple indices, allows to ignore missing ones, default u'none'

- **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)

**Returns** resulting JSON

See also:


**get_aliases** *(index=None, name=None, *, local=default, timeout=default)*

A coroutine that retrieves a specified aliases.

**Parameters**

- **index** – A comma-separated list of index names to filter aliases

- **name** – A comma-separated list of alias names to filter

- **local** – Return local information, do not retrieve the state from master node (default: false)

- **timeout** – Explicit operation timeout

**Returns** resulting JSON

See also:


**update_aliases** *(body, *, timeout=default, master_timeout=default)*

A coroutine that updates a specified aliases.

**Parameters**

- **body** – The definition of actions to perform

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

- **timeout** – Request timeout

**Returns** resulting JSON

See also:


**delete_alias** *(index, name, *, timeout=default, master_timeout=default)*

A coroutine that deletes a specified aliases.

**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

**Returns** resulting JSON

See also:

**put_template**(name, body, *, create=default, order=default, timeout=default, master_timeout=default, flat_settings=default)
A coroutine that creates an index template that will be automatically applied to new indices created.

**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
• **order** – The order for this template when merging multiple matching ones (higher numbers are merged later, overriding the lower numbers)
• **master_timeout** – Specify timeout for connection to master
• **timeout** – Explicit operation timeout
• **flat_settings** – Return settings in flat format (default: false)

**Returns** resulting JSON

See also:

**exists_template**(name, *, local=default)
A coroutine that returns a boolean indicating whether given template exists.

**Parameters**
• **name** – The name of the template
• **local** – Return local information, do not retrieve the state from master node (default: false)

**Returns** resulting JSON

See also:

**get_template**(name=None, *, flat_settings=default, local=default)
A coroutine that retrieves an index template by its name.

**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)

**Returns** resulting JSON
delete_template (name, *, timeout=default, master_timeout=default)

A coroutine that deletes an index template by its name.

Parameters

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

Returns resulting JSON

See also:


put_warmer (name, body, index=None, doc_type=None, *, allow_no_indices=default, expand_wildcards=default, ignore_unavailable=default, master_timeout=default)

A coroutine that create an index warmer to run registered search requests to warm up the index before it is available for search.

Parameters

- **name** – The name of the warmer
- **body** – The search request definition for the warmer (query, filters, facets, sorting, etc)
- **index** – A comma-separated list of index names to register the warmer for; use _all or omit to perform the operation on all indices
- **doc_type** – A comma-separated list of document types to register the warmer for; leave empty to perform the operation on all types
- **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices in the search request to warm. (This includes _all string or when no indices have been specified)
- **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both, in the search request to warm., default u'open'
- **ignore_unavailable** – Whether specified concrete indices should be ignored when unavailable (missing or closed) in the search request to warm
- **master_timeout** – Specify timeout for connection to master

Returns resulting JSON

See also:


get_warmer (index=None, doc_type=None, name=None, *, allow_no_indices=default, expand_wildcards=default, ignore_unavailable=default, local=default)

A coroutine that retrieves an index warmer.

Parameters

- **index** – A comma-separated list of index names to restrict the operation; use _all to perform the operation on all indices
- **doc_type** – A comma-separated list of document types to restrict the operation; leave empty to perform the operation on all types
• **name** – The name of the warmer (supports wildcards); leave empty to get all warmers

• **allow_no_indices** – Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes _all string or when no indices have been specified)

• **expand_wildcards** – Whether to expand wildcard expression to concrete indices that are open, closed or both. default u’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)

  Returns resulting JSON

  See also:


**delete_warmer**(index, name, *, master_timeout=default)

A coroutine that deletes an index warmer.

Parameters

• **index** – A comma-separated list of index names to delete warmers from (supports wildcards); use _all to perform the operation on all indices.

• **name** – A comma-separated list of warmer names to delete (supports wildcards); use _all to delete all warmers in the specified indices.

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

  Returns resulting JSON

  See also:


**CatClient**

class aioes.client.CatClient

Class for retrieving elasticsearch information in human-readable way.

**aliases**(*, name=default, h=default, help=default, local=default, master_timeout=default, v=default)

A coroutine that returns an info about aliases.

Parameters

• **name** – A comma-separated list of alias names to return

• **h** – Comma-separated list of column names to display

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

• **local** – Return local information, do not retrieve the state from master node (default: false)

• **master_timeout** – Explicit operation timeout for connection to master node

• **v** – Verbos mode. Display column headers, default False

  Returns resulting text
allocation (node_id=None, *, h=default, help=default, local=default, master_timeout=default, v=default)
A coroutine that returns a snapshot of how shards have located around the cluster and the state of disk usage.

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

Returns resulting text

See also:

count (index=None, *, h=default, help=default, local=default, master_timeout=default, v=default)
A coroutine that returns an info about aliases.

Parameters

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

Returns resulting text

See also:

health (*, h=default, help=default, local=default, master_timeout=default, ts=default, v=default)
A coroutine that returns a health, which is a terse, one-line representation of the same information from health() API

Parameters

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

**ts** – Set to false to disable timestamping, default True

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

**Returns** resulting text

**See also:**


**help** (*, help=default)

A coroutine that returns help banner.

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

**Returns** help text

**See also:**


**indices**(self, index=None, *, bytes=default, h=default, help=default, local=default, master_timeout=default, pri=default, v=default)

A coroutine that returns a cross-section of each index

**Parameters**

• **index** – A comma-separated list of index names to limit the returned information

• **bytes** – The unit in which to display byte values

• **h** – Comma-separated list of column names to display

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

• **local** – Return local information, do not retrieve the state from master node (default: false)

• **master_timeout** – Explicit operation timeout for connection to master node

• **pri** – Set to true to return stats only for primary shards, default False

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

**Returns** resulting text

**See also:**


**master**(*, h=default, help=default, local=default, master_timeout=default, v=default)

A coroutine that displays the master’s node ID, bound IP address, and node name

**Parameters**

• **h** – Comma-separated list of column names to display

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

• **local** – Return local information, do not retrieve the state from master node (default: false)

• **master_timeout** – Explicit operation timeout for connection to master node

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

**Returns** resulting text
nodes (*, h=default, help=default, local=default, master_timeout=default, v=default)
A coroutine that shows the cluster topology.

Parameters

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

Returns resulting text

See also:

recovery (index=\texttt{None}, *, bytes=default, h=default, help=default, local=default, master_timeout=default, v=default)
A coroutine that shows the view of shard replication.

Parameters

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

Returns resulting text

See also:

shards (index=\texttt{None}, *, h=default, help=default, local=default, master_timeout=default, v=default)
A coroutine that shows the detailed view of what nodes contain which shards.

Parameters

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

See also:
Returns resulting text

See also:


segments (index=None, *, h=default, help=default, local=default, master_timeout=default, v=default)
A coroutine that shows the detailed view of Lucene segments per index

Parameters

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

Returns resulting text

See also:


pending_tasks (*, h=default, help=default, local=default, master_timeout=default, v=default)
A coroutine that provides the same information as the pending_tasks() API in a convenient tabular format.

Parameters

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

Returns resulting text

See also:


thread_pool (*, full_id=default, h=default, help=default, local=default, master_timeout=default, v=default)
A coroutine that provides information about thread pools.

Parameters

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

**Returns** resulting text

**See also:**

**fielddata** (*, fields=default, bytes=default, h=default, help=default, local=default, master_timeout=default, v=default)  
A coroutine that provides information about currently loaded fielddata on a per-node basis

**Parameters**

• **fields** – A comma-separated list of fields to return the fielddata size
• **bytes** – The unit in which to display byte values
• **h** – Comma-separated list of column names to display
• **help** – Return help information (default: ‘false’)
• **local** – Return local information, do not retrieve the state from master node (default: false)
• **master_timeout** – Explicit operation timeout for master connection node
• **v** – Verbose mode. Display column headers (default: ‘false’)

**Returns** resulting text

**See also:**

**plugins** (*, h=default, help=default, local=default, master_timeout=default, v=default)  
A coroutine that provides information about plugins.

**Parameters**

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

**Returns** resulting text

**See also:**

**ClusterClient**

class **aioes.client.ClusterClient**  
Class for retrieving elasticsearch information in human-readable way.
**health** (index=None, *, level=default, local=default, master_timeout=default, timeout=default, wait_for_active_shards=default, wait_for_nodes=default, wait_for_relocating_shards=default, wait_for_status=default)

A coroutine that returns a very simple status on the health of the cluster.

**Parameters**

- **index** – Limit the information returned to a specific index
- **level** – Specify the level of detail for returned information, default u’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_nodes** – Wait until the specified number of nodes is available
- **wait_for_relocating_shards** – Wait until the specified number of relocating shards is finished
- **wait_for_status** – Wait until cluster is in a specific state, default None

**Returns** resulting text

**See also:**


**pending_tasks** (*, local=default, master_timeout=default)

A coroutine that returns a list of any cluster-level changes (e.g. create index, update mapping, allocate or fail shard) which have not yet been executed.

**Parameters**

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

**Returns** resulting text

**See also:**


**state** (metric=None, index=None, *, index_templates=default, local=default, master_timeout=default, flat_settings=default)

A coroutine that returns a comprehensive state information of the whole cluster.

**Parameters**

- **metric** – Limit the information returned to the specified metrics. Possible values: “_all”, “blocks”, “index_templates”, “metadata”, “nodes”, “routing_table”, “master_node”, “version”
- **index** – A comma-separated list of index names; use _all or empty string to perform the operation on all indices
- **index_templates** – A comma separated list to return specific index templates when returning metadata.
- **local** – Return local information, do not retrieve the state from master node (default: false)
- **master_timeout** – Specify timeout for connection to master
- **flat_settings** – Return settings in flat format (default: false)

**Returns** resulting text

See also:


**stats** *(node_id=None, *, flat_settings=default, human=default)*

A coroutine that returns statistics from a cluster wide perspective. The API returns basic index metrics and information about the current nodes that form the cluster.

**Parameters**

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

**Returns** resulting text

See also:


**reroute** *(body=None, *, dry_run=default, explain=default, filter_metadata=default, master_timeout=default, timeout=default)*

A coroutine that executes a cluster reroute allocation command including specific commands.

**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
- **filter_metadata** – Don’t return cluster state metadata (default: false)
- **master_timeout** – Explicit operation timeout for connection to master node
- **timeout** – Explicit operation timeout

**Returns** resulting text

See also:


**get_settings** *(*, flat_settings=default, master_timeout=default, timeout=default)*

A coroutine that returns cluster settings.

**Parameters**

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

**Returns** resulting text
See also:


put_settings(body, *, flat_settings=default)
A coroutine that updates cluster settings.

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)

Returns resulting text

See also:


NodesClient

class aioes.client.NodesClient
Class for getting information about elasticsearch nodes.

info(node_id=None, metric=None, *, flat_settings=default, human=default)
A coroutine that retrieves one or more (or all) of the cluster nodes information.

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. Choices are "settings", "os", "process", "jvm", "thread_pool", "network", "transport", "http", "plugin"
- **flat_settings** – Return settings in flat format (default: False)
- **human** – Whether to return time and byte values in human-readable format, default False

Returns resulting info

See also:


shutdown(node_id=None, *, delay=default, exit=default)
A coroutine that shutdowns one or more (or all) nodes in the cluster.

Parameters

- **node_id** – A comma-separated list of node IDs or names to perform the operation on; use _local to perform the operation on the node you’re connected to, leave empty to perform the operation on all nodes
- **delay** – Set the delay for the operation (default: 1s)
- **exit** – Exit the JVM as well (default: true)

Returns resulting info
See also:


stats(node_id=None, metric=None, index_metric=None, *, completion_fields=default, field_data_fields=default, fields=default, groups=default, human=default, level=default, types=default)

A coroutine that allows to retrieve one or more (or all) of the cluster nodes statistics.

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. Possible options are: “_all”, “breaker”, “fs”, “http”, “indices”, “jvm”, “network”, “os”, “process”, “thread_pool”, “transport”
- **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. Possible options are: “_all”, “completion”, “docs”, “fielddata”, “filter_cache”, “flush”, “get”, “id_cache”, “indexing”, “merge”, “percolate”, “refresh”, “search”, “segments”, “store”, “warmer”
- **completion_fields** – A comma-separated list of fields for fielddata and suggest index metric (supports wildcards)
- **fielddata_fields** – A comma-separated list of fields for fielddata index metric (supports wildcards)
- **fields** – A comma-separated list of fields for fielddata and completion index metric (supports wildcards)
- **groups** – A comma-separated list of search groups for search index metric
- **human** – Whether to return time and byte values in human-readable format., default False
- **level** – Return indices stats aggregated at node, index or shard level, default ‘node’
- **types** – A comma-separated list of document types for the indexing index metric

Returns resulting info

See also:


hot_threads(node_id=None, *, type_=default, interval=default, snapshots=default, threads=default)

A coroutine that allows to get the current hot threads on each node in the cluster.

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
- **type** – The type to sample (default: cpu)
- **interval** – The interval for the second sampling of threads
- **snapshots** – Interval for the second sampling of threads (default: 10)
- **threads** – Number of samples of thread stacktrace (default: 3)

Returns resulting info
See also:

**SnapshotClient**

class *aioes.client.SnapshotClient*

Class for manipulating elasticsearch snapshots.

**create** *(repository, snapshot, body=None, *, master_timeout=default, wait_for_completion=default)*

A coroutine that creates a snapshot in repository

**Parameters**

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

**Returns** resulting info.

See also:

**delete** *(repository, snapshot, *, master_timeout=default)*

A coroutine that deletes a snapshot in repository

**Parameters**

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

**Returns** resulting info.

See also:

**get** *(repository, snapshot, *, master_timeout=default)*

A coroutine that retrieves information about a snapshot.

**Parameters**

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

**Returns** resulting info.

See also:

**delete_repository** *(repository, *, master_timeout=default, timeout=default)*

A coroutine that removes a shared file system repository.
Parameters

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

Returns resulting info.

See also:


get_repository(repository=None, *, local=default, master_timeout=default)
A coroutine that returns information about registered repositories.

Parameters

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

Returns resulting info.

See also:


create_repository(repository, body, *, master_timeout=default, timeout=default)
A coroutine that registers a shared file system repository.

Parameters

- repository – A repository name
- body – The repository definition
- master_timeout – Explicit operation timeout for connection to master node
- timeout – Explicit operation timeout

Returns resulting info.

See also:


restore(repository, snapshot, body=None, *, master_timeout=default, wait_for_completion=default)
A coroutine that restores a snapshot.

Parameters

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

Returns resulting info.
See also:

status (repository=None, snapshot=None, *, master_timeout=default)
A coroutine that returns snapshot status info.

Parameters

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

Returns resulting snapshot info.

See also:
_snapshot_status

Glossary

aiohttp asyncio compatible library for working with HTTP protocol.

asyncio Reference implementation of PEP 3156 — Asynchronous IO Support Rebooted.

ElasticSearch A search server based on Lucene. It provides a distributed, multitenant-capable full-text search engine with a RESTful web interface and schema-free JSON documents.

http://www.elasticsearch.org

elasticsearch-py Official python client for ElasticSearch

ipdb ipdb exports functions to access the IPython debugger, which features tab completion, syntax highlighting, better tracebacks, better introspection with the same interface as the pdb module.

pep8 Python style guide checker

pep8 is a tool to check your Python code against some of the style conventions in PEP 8 – Style Guide for Python Code.

pyflakes passive checker of Python programs

A simple program which checks Python source files for errors.

Pyflakes analyzes programs and detects various errors. It works by parsing the source file, not importing it, so it is safe to use on modules with side effects. It’s also much faster.

https://pypi.python.org/pypi/pyflakes

venv standard python module for creating lightweight “virtual environments” with their own site directories, optionally isolated from system site directories. Each virtual environment has its own Python binary (allowing creation of environments with various Python versions) and can have its own independent set of installed Python packages in its site directories.

https://docs.python.org/dev/library/venv.html

virtualenv The tool to create isolated Python environments. It’s not included into python stdlib’s but it is very popular instrument.

venv and virtualenv does almost the same, venv has been developed after the virtualenv.

http://virtualenv.readthedocs.org/en/latest/
**virtualenvwrapper**  virtualenvwrapper is a set of extensions to Ian Bicking's `virtualenv` tool. The extensions include wrappers for creating and deleting virtual environments and otherwise managing your development workflow, making it easier to work on more than one project at a time without introducing conflicts in their dependencies.

virtualenvwrapper is my choice, highly recommend the tool to everyone.

http://virtualenvwrapper.readthedocs.org/en/latest/
Indices and tables

- genindex
- modindex
- search
a
  aioes, 11
Index

A
aioes (module), 11
aioes.client.CatClient (class in aioes), 43
aioes.client.ClusterClient (class in aioes), 48
aioes.client.IndicesClient (class in aioes), 28
aioes.client.NodesClient (class in aioes), 51
aioes.client.SnapshotClient (class in aioes), 53
aiohttp, 55
aliases() (aioes.aioes.client.CatClient method), 43
allocation() (aioes.aioes.client.CatClient method), 44
analyze() (aioes.aioes.client.IndicesClient method), 28
asyncio, 55

B
benchmark() (aioes.Elasticsearch method), 27
bulk() (aioes.Elasticsearch method), 20

C
clear_cache() (aioes.aioes.client.IndicesClient method), 36
clear_scroll() (aioes.Elasticsearch method), 19
close() (aioes.aioes.client.IndicesClient method), 29
count() (aioes.aioes.client.CatClient method), 44
count() (aioes.Elasticsearch method), 20
count_percolate() (aioes.Elasticsearch method), 24
create() (aioes.aioes.client.IndicesClient method), 29
create() (aioes.aioes.client.SnapshotClient method), 53
create() (aioes.Elasticsearch method), 12
create_repository() (aioes.aioes.client.SnapshotClient method), 54

d
delete() (aioes.aioes.client.IndicesClient method), 30
delete() (aioes.aioes.client.SnapshotClient method), 53
delete() (aioes.Elasticsearch method), 19
delete_alias() (aioes.aioes.client.IndicesClient method), 40
delete_by_query() (aioes.Elasticsearch method), 21
delete_mapping() (aioes.aioes.client.IndicesClient method), 38
delete_repository() (aioes.aioes.client.SnapshotClient method), 53
delete_script() (aioes.Elasticsearch method), 27
delete_template() (aioes.aioes.client.IndicesClient method), 42
delete_template() (aioes.Elasticsearch method), 28
delete_warmer() (aioes.aioes.client.IndicesClient method), 43

E
ElasticSearch, 55
Elasticsearch (class in aioes), 11
elasticsearch-py, 55
exists() (aioes.aioes.client.IndicesClient method), 31
exists() (aioes.Elasticsearch method), 13
exists_alias() (aioes.aioes.client.IndicesClient method), 39
exists_template() (aioes.aioes.client.IndicesClient method), 41
exists_type() (aioes.aioes.client.IndicesClient method), 31
explain() (aioes.Elasticsearch method), 18

F
fielddata() (aioes.aioes.client.CatClient method), 48
flush() (aioes.aioes.client.IndicesClient method), 30

G
get() (aioes.aioes.client.SnapshotClient method), 53
get() (aioes.Elasticsearch method), 13
get_alias() (aioes.aioes.client.IndicesClient method), 39
get_aliases() (aioes.aioes.client.IndicesClient method), 40
get_field_mapping() (aioes.aioes.client.IndicesClient method), 38
get_mapping() (aioes.aioes.client.IndicesClient method), 37
get_repository() (aioes.aioes.client.SnapshotClient method), 54
get_script() (aioes.Elasticsearch method), 27
get_settings() (aioes.aioes.client.ClusterClient method), 50
get_settings() (aioes.aioes.client.IndicesClient method), 32
get_source() (aioes.Elasticsearch method), 14
get_template() (aioes.aioes.client.IndicesClient method), 41
get_template() (aioes.Elasticsearch method), 28
get_warmer() (aioes.aioes.client.IndicesClient method), 42
health() (aioes.aioes.client.CatClient method), 44
health() (aioes.aioes.client.ClusterClient method), 48
help() (aioes.aioes.client.CatClient method), 45
hot_threads() (aioes.aioes.client.NodesClient method), 52
index() (aioes.Elasticsearch method), 11
indices() (aioes.aioes.client.CatClient method), 45
info() (aioes.aioes.client.NodesClient method), 51
info() (aioes.Elasticsearch method), 11
ipdb, 55
master() (aioes.aioes.client.CatClient method), 45
mget() (aioes.Elasticsearch method), 14
mlt() (aioes.Elasticsearch method), 24
msearch() (aioes.Elasticsearch method), 23
mtermvectors() (aioes.Elasticsearch method), 26
nodes() (aioes.aioes.client.CatClient method), 46
open() (aioes.aioes.client.IndicesClient method), 29
optimize() (aioes.aioes.client.IndicesClient method), 34
pending_tasks() (aioes.aioes.client.CatClient method), 47
pending_tasks() (aioes.aioes.client.ClusterClient method), 49
pep8, 55
percolate() (aioes.Elasticsearch method), 22
ping() (aioes.Elasticsearch method), 11
plugins() (aioes.aioes.client.CatClient method), 48
put_alias() (aioes.aioes.client.IndicesClient method), 39
put_mapping() (aioes.aioes.client.IndicesClient method), 37
put_script() (aioes.Elasticsearch method), 27
put_settings() (aioes.aioes.client.IndicesClient method), 27
put_template() (aioes.aioes.client.IndicesClient method), 41
put_template() (aioes.Elasticsearch method), 28
put_warmer() (aioes.aioes.client.IndicesClient method), 42
pyflakes, 55
Python Enhancement Proposals
PEP 3156, 55
PEP 8, 55
recovery() (aioes.aioes.client.CatClient method), 46
recovery() (aioes.aioes.client.IndicesClient method), 35
refresh() (aioes.aioes.client.IndicesClient method), 34
reroute() (aioes.aioes.client.ClusterClient method), 50
restore() (aioes.aioes.client.SnapshotClient method), 54
scroll() (aioes.Elasticsearch method), 19
search() (aioes.Elasticsearch method), 15
search_shards() (aioes.Elasticsearch method), 17
search_template() (aioes.Elasticsearch method), 17
segments() (aioes.aioes.client.CatClient method), 47
segments() (aioes.aioes.client.IndicesClient method), 34
shards() (aioes.aioes.client.ClusterClient method), 46
shutdown() (aioes.aioes.client.NodesClient method), 51
state() (aioes.aioes.client.ClusterClient method), 49
stats() (aioes.aioes.client.ClusterClient method), 49
stats() (aioes.aioes.client.IndicesClient method), 33
stats() (aioes.aioes.client.NodesClient method), 52
status() (aioes.aioes.client.IndicesClient method), 33
status() (aioes.aioes.client.SnapshotClient method), 55
suggest() (aioes.Elasticsearch method), 22
termvector() (aioes.Elasticsearch method), 25
thread_pool() (aioes.aioes.client.CatClient method), 47
update() (aioes.Elasticsearch method), 15
update_aliases() (aioes.aioes.client.IndicesClient method), 40
virtualenv, 55
virtualenvwrapper, 56