* [Preface](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-api.html)
* [Javadoc](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/_javadoc.html)
* [Maven Repository](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/_maven_repository.html)
* [Dealing with JAR dependency conflicts](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/_dealing_with_jar_dependency_conflicts.html)
* [Embedding jar with dependencies](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/_embedding_jar_with_dependencies.html)
* [Deploying in JBoss EAP6 module](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/_deploying_in_jboss_eap6_module.html)
* [Client](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/client.html)
* [Document APIs](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs.html)
* [Search API](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-search.html)
* [Aggregations](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-aggs.html)
* [Query DSL](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-query-dsl.html)
* [Indexed Scripts API](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/indexed-scripts.html)
* [Java API Administration](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin.html)

Preface[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/index.asciidoc)

This section describes the Java API that elasticsearch provides. All elasticsearch operations are executed using a [Client](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/client.html) object. All operations are completely asynchronous（异步操作） in nature (either accepts a listener, or returns a future).

Additionally, operations on a client may be accumulated and executed in [Bulk](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-bulk.html)（散装，类似批量操作）.

Note, all the APIs are exposed through the Java API (actually, the Java API is used internally to execute them).

Starting from version 5.6.0, a new Java client has been released: the [Java High Level REST Client](https://www.elastic.co/guide/en/elasticsearch/client/java-rest/5.6/java-rest-high.html). This new client is designed to replace the TransportClient in Java applications which will be deprecated in future versions of Elasticsearch.

[Maven Repository](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/_maven_repository.html)

* [Log4j 2 Logger](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/_log4j_2_logger.html)
* [Using another Logger](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/_using_another_logger.html)

Maven Repository[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/index.asciidoc)

Elasticsearch is hosted on [Maven Central](http://search.maven.org/#search%7Cga%7C1%7Ca%3A%22elasticsearch%22).

For example, you can define the latest version in your pom.xml file:

<dependency>

<groupId>org.elasticsearch.client</groupId>

<artifactId>transport</artifactId>

<version>5.6.1</version>

</dependency>

## Log4j 2 Logger[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/index.asciidoc)

You need to also include Log4j 2 dependencies:

<dependency>

<groupId>org.apache.logging.log4j</groupId>

<artifactId>log4j-core</artifactId>

<version>2.9.1</version>

</dependency>

And also provide a Log4j 2 configuration file in your classpath. For example, you can add in your src/main/resources project dir a log4j2.properties file like:

appender.console.type = Console

appender.console.name = console

appender.console.layout.type = PatternLayout

rootLogger.level = info

rootLogger.appenderRef.console.ref = console

## Using another Logger[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/index.asciidoc)

If you want to use another logger than Log4j 2, you can use [SLF4J](http://www.slf4j.org/) bridge to do that:

<dependency>

<groupId>org.apache.logging.log4j</groupId>

<artifactId>log4j-to-slf4j</artifactId>

<version>2.9.1</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.7.24</version>

</dependency>

[This page](http://www.slf4j.org/manual.html) lists implementations you can use. Pick your favorite logger and add it as a dependency. As an example, we will use the slf4j-simple logger:

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-simple</artifactId>

<version>1.7.21</version>

</dependency>

Dealing with JAR dependency conflicts[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/index.asciidoc)

If you want to use Elasticsearch in your Java application, you may have to deal with version conflicts with third party dependencies like Guava and Joda. For instance, perhaps Elasticsearch uses Joda 2.8, while your code uses Joda 2.1.

You have two choices:

* The simplest solution is to upgrade. Newer module versions are likely to have fixed old bugs. The further behind you fall, the harder it will be to upgrade later. Of course, it is possible that you are using a third party dependency that in turn depends on an outdated version of a package, which prevents you from upgrading.
* The second option is to relocate the troublesome dependencies and to shade them either with your own application or with Elasticsearch and any plugins needed by the Elasticsearch client.

The ["To shade or not to shade" blog post](https://www.elastic.co/blog/to-shade-or-not-to-shade) describes all the steps for doing so.

# Embedding jar with dependencies[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/index.asciidoc)

If you want to create a single jar containing your application and all dependencies, you should not use maven-assembly-plugin for that because it can not deal with META-INF/services structure which is required by Lucene jars.

Instead, you can use maven-shade-plugin and configure it as follow:

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-shade-plugin</artifactId>

<version>2.4.1</version>

<executions>

<execution>

<phase>package</phase>

<goals><goal>shade</goal></goals>

<configuration>

<transformers>

<transformer implementation="org.apache.maven.plugins.shade.resource.ServicesResourceTransformer"/>

</transformers>

</configuration>

</execution>

</executions>

</plugin>

Note that if you have a main class you want to automatically call when running java -jar yourjar.jar, just add it to the transformers:

<transformer implementation="org.apache.maven.plugins.shade.resource.ManifestResourceTransformer">

<mainClass>org.elasticsearch.demo.Generate</mainClass>

</transformer>

# Deploying in JBoss EAP6 module[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/index.asciidoc)

Elasticsearch and Lucene classes need to be in the same JBoss module.

You should define a module.xml file like this:

<?xml version="1.0" encoding="UTF-8"?>

<module name="org.elasticsearch">

<resources>

<!-- Elasticsearch -->

<resource-root path="elasticsearch-2.0.0.jar"/>

<!-- Lucene -->

<resource-root path="lucene-core-5.1.0.jar"/>

<resource-root path="lucene-analyzers-common-5.1.0.jar"/>

<resource-root path="lucene-queries-5.1.0.jar"/>

<resource-root path="lucene-memory-5.1.0.jar"/>

<resource-root path="lucene-highlighter-5.1.0.jar"/>

<resource-root path="lucene-queryparser-5.1.0.jar"/>

<resource-root path="lucene-sandbox-5.1.0.jar"/>

<resource-root path="lucene-suggest-5.1.0.jar"/>

<resource-root path="lucene-misc-5.1.0.jar"/>

<resource-root path="lucene-join-5.1.0.jar"/>

<resource-root path="lucene-grouping-5.1.0.jar"/>

<resource-root path="lucene-spatial-5.1.0.jar"/>

<resource-root path="lucene-expressions-5.1.0.jar"/>

<!-- Insert other resources here -->

</resources>

<dependencies>

<module name="sun.jdk" export="true" >

<imports>

<include path="sun/misc/Unsafe" />

</imports>

</module>

<module name="org.apache.log4j"/>

<module name="org.apache.commons.logging"/>

<module name="javax.api"/>

</dependencies>

</module>

[Client](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/client.html)

* [**Transport Client**](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/transport-client.html)
* [Connecting a Client to a Coordinating Only Node](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/client-connected-to-client-node.html)

Client[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/client.asciidoc)

You can use the **Java client** in multiple ways:

* Perform standard [index](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-index.html), [get](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-get.html), [delete](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-delete.html) and [search](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-search.html) operations on an existing cluster
* Perform administrative tasks（执行管理任务） on a running cluster

Obtaining an elasticsearch Client is simple. The most common way to get a client is by creating a [TransportClient](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/transport-client.html" \o "Transport Client) that connects to a cluster.

Important 版本尽量一致或至少兼容，2.x和5.x差别较大；java client和集群中node的版本尽量相同，至少兼容

The client must have the same major version (e.g. 2.x, or 5.x) as the nodes in the cluster. Clients may connect to clusters which have a different minor version (e.g. 2.3.x) but it is possible that new functionality may not be supported. Ideally, the client should have the same version as the cluster.

Warning TransportClient 将要被废弃了，代之以High Level REST Client来处理http请求而不是序列化java请求（TransportClient 目前的实现是这样）

The TransportClient is aimed to be replaced by the Java High Level REST Client, which executes HTTP requests instead of serialized Java requests. The TransportClient will be deprecated in upcoming versions of Elasticsearch and it is advised to use the Java High Level REST Client instead.

## Transport Client[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/client.asciidoc)

The TransportClient connects remotely to an Elasticsearch cluster using the transport module. It does not join the cluster（这种连接并不加入集群，即不作为集群的一个client，仅仅是为了连接集群处理操作）, but simply gets one or more initial transport addresses and communicates with them in round robin fashion（轮询） on each action (though most actions will probably be "two hop" （P2P网络的一种搜索协议）operations).

// on startup

TransportClient client = new PreBuiltTransportClient(Settings.EMPTY)

.addTransportAddress(new InetSocketTransportAddress(InetAddress.getByName("host1"), 9300))

.addTransportAddress(new InetSocketTransportAddress(InetAddress.getByName("host2"), 9300));

// on shutdown

client.close();

Note that you have to set the cluster name if you use one different than "elasticsearch":

Settings settings = Settings.builder()

.put("cluster.name", "myClusterName").build();

TransportClient client = new PreBuiltTransportClient(settings);

//Add transport addresses and do something with the client...

The Transport client comes with a cluster sniffing feature（自动发现功能，即连接上一个node后自动发现其他的节点） which allows it to dynamically add new hosts and remove old ones. When sniffing is enabled, the transport client will connect to the nodes in its internal node list, which is built via calls to addTransportAddress. After this, the client will call the internal cluster state API on those nodes to discover available data nodes. The internal node list of the client will be replaced with those data nodes only. This list is refreshed every five seconds by default. Note that the IP addresses the sniffer connects to are the ones declared as the publishaddress in those node’s elasticsearch config（连接的带有嗅探功能的节点需要在配置文件中配置为publish address）.

Keep in mind that the list might possibly not include the original node it connected to if that node is not a data node. If, for instance, you initially connect to a master node, after sniffing, no further requests will go to that master node, but rather to any data nodes instead. The reason the transport client excludes non-data nodes is to avoid sending search traffic to master only nodes（变给非data node造成搜索压力，该client不会请求master节点查询数据）.

In order to enable sniffing, set client.transport.sniff to true:

Settings settings = Settings.builder()

.put("client.transport.sniff", true).build();

TransportClient client = new PreBuiltTransportClient(settings);

Other transport client level settings include:

| **Parameter** | **Description** |
| --- | --- |
| client.transport.ignore\_cluster\_name | Set to true to ignore cluster name validation of connected nodes（不验证集群名）. (since 0.19.4) |
| client.transport.ping\_timeout | The time to wait for a ping response from a node. Defaults to 5s. |
| client.transport.nodes\_sampler\_interval | How often to sample / ping the nodes listed and connected. Defaults to 5s. |

## Connecting a Client to a Coordinating Only Node[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/client.asciidoc)

You can start locally a [Coordinating Only Node](https://www.elastic.co/guide/en/elasticsearch/reference/5.6/modules-node.html#coordinating-only-node) and then simply create a [TransportClient](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/transport-client.html" \o "Transport Client) in your application which connects to this Coordinating Only Node.

This way, the coordinating only node will be able to load whatever plugin you need (think about discovery plugins for example).

[Document APIs](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs.html)

* [Index API](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-index.html)
* [Get API](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-get.html)
* [Delete API](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-delete.html)
* [Delete By Query API](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-delete-by-query.html)
* [Update API](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-update.html)
* [Multi Get API](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-multi-get.html)
* [Bulk API](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-bulk.html)
* [Using Bulk Processor](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-bulk-processor.html)

## Index API[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/index_.asciidoc)

The index API allows one to index a typed JSON document into a specific index and make it searchable.

### Generate JSON document[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/index_.asciidoc)

There are several different ways of generating a JSON document:

* Manually (aka do it yourself) using native byte[] or as a String
* Using a Map that will be automatically converted to its JSON equivalent
* Using a third party library to serialize your beans such as [Jackson](http://wiki.fasterxml.com/JacksonHome)
* Using built-in helpers XContentFactory.jsonBuilder()内置的json构建工厂

Internally, each type is converted to byte[] （所有的这些json最后都被转换为byte[]）(so a String is converted to a byte[]). Therefore, if the object is in this form already, then use it. The jsonBuilder is highly optimized JSON generator（最好使用内置的jsonbuilder，性能做了优化） that directly constructs a byte[].

#### Do It Yourself[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/index_.asciidoc" \o "Edit this page on GitHub)

Nothing really difficult here but note that you will have to encode dates(编码时间为一个时间格式) according to the [Date Format](https://www.elastic.co/guide/en/elasticsearch/reference/5.6/mapping-date-format.html).

String json = "{" +

"\"user\":\"kimchy\"," +

"\"postDate\":\"2013-01-30\"," +

"\"message\":\"trying out Elasticsearch\"" +

"}";

#### Using Map[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/index_.asciidoc)

Map is a key:values pair collection. It represents a JSON structure:

Map<String, Object> json = new HashMap<String, Object>();

json.put("user","kimchy");

json.put("postDate",new Date());

json.put("message","trying out Elasticsearch");

#### Serialize your beans[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/index_.asciidoc)

You can use [Jackson](http://wiki.fasterxml.com/JacksonHome) to serialize your beans to JSON. Please add [Jackson Databind](http://search.maven.org/#search%7Cga%7C1%7Cjackson-databind) to your project. Then you can use ObjectMapper to serialize your beans:

import com.fasterxml.jackson.databind.\*;

// instance a json mapper

ObjectMapper mapper = new ObjectMapper(); // create once, reuse可以重复使用，应该放在外面避免生成多个实例

// generate json

byte[] json = mapper.writeValueAsBytes(yourbeaninstance);

#### Use Elasticsearch helpers[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/index_.asciidoc)

Elasticsearch provides built-in helpers（内置的jsonbuilder，性能更好） to generate JSON content.

import static org.elasticsearch.common.xcontent.XContentFactory.\*;

XContentBuilder builder = jsonBuilder()

.startObject()

.field("user", "kimchy")

.field("postDate", new Date())

.field("message", "trying out Elasticsearch")

.endObject()

Note that you can also add arrays with startArray(String) and endArray() methods. By the way, the field method accepts many object types. You can directly pass numbers, dates and even other XContentBuilder objects.

If you need to see the generated JSON content, you can use the string() method.

String json = builder.string();

### Index document[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/index_.asciidoc)

The following example indexes a JSON document into an index called twitter, under a type called tweet, with id valued 1:

import static org.elasticsearch.common.xcontent.XContentFactory.\*;

IndexResponse response = client.prepareIndex("twitter", "tweet", "1")

.setSource(jsonBuilder()

.startObject()

.field("user", "kimchy")

.field("postDate", new Date())

.field("message", "trying out Elasticsearch")

.endObject()

)

.get();

Note that you can also index your documents as JSON String and that you don’t have to give an ID:

String json = "{" +

"\"user\":\"kimchy\"," +

"\"postDate\":\"2013-01-30\"," +

"\"message\":\"trying out Elasticsearch\"" +

"}";

IndexResponse response = client.prepareIndex("twitter", "tweet")

      .setSource(json, XContentType.JSON)

.get();

IndexResponse object will give you a report:

// Index name

String \_index = response.getIndex();

// Type name

String \_type = response.getType();

// Document ID (generated or not)

String \_id = response.getId();

// Version (if it's the first time you index this document, you will get: 1)

long \_version = response.getVersion();

// status has stored current instance statement.

RestStatus status = response.status();

For more information on the index operation, check out the REST [index](https://www.elastic.co/guide/en/elasticsearch/reference/5.6/docs-index_.html) docs.

### Operation Threading[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/index_.asciidoc" \o "Edit this page on GitHub)

The index API allows one to set the threading model the operation will be performed when the actual execution of the API is performed on the same node (the API is executed on a shard that is allocated on the same server).

The options are to execute the operation on a different thread, or to execute it on the calling thread (note that the API is still asynchronous). By default, operationThreaded is set to true which means the operation is executed on a different thread.

## Get API[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/get.asciidoc)

The get API allows to get a typed JSON document from the index based on its id. The following example gets a JSON document from an index called twitter, under a type called tweet, with id valued 1:

GetResponse response = client.prepareGet("twitter", "tweet", "1").get();

For more information on the get operation, check out the REST [get](https://www.elastic.co/guide/en/elasticsearch/reference/5.6/docs-get.html) docs.

### Operation Threading[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/get.asciidoc" \o "Edit this page on GitHub)

The get API allows to set the threading model the operation will be performed when the actual execution of the API is performed on the same node (the API is executed on a shard that is allocated on the same server).

The options are to execute the operation on a different thread, or to execute it on the calling thread (note that the API is still async). By default, operationThreaded is set to true which means the operation is executed on a different thread. Here is an example that sets it to false:

GetResponse response = client.prepareGet("twitter", "tweet", "1")

.setOperationThreaded(false)

.get();

## Delete API[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/delete.asciidoc)

The delete API allows one to delete a typed JSON document from a specific index based on its id. The following example deletes the JSON document from an index called twitter, under a type called tweet, with id valued 1:

DeleteResponse response = client.prepareDelete("twitter", "tweet", "1").get();

For more information on the delete operation, check out the [delete API](https://www.elastic.co/guide/en/elasticsearch/reference/5.6/docs-delete.html) docs.

### Operation Threading[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/delete.asciidoc" \o "Edit this page on GitHub)

The delete API allows to set the threading model the operation will be performed when the actual execution of the API is performed on the same node (the API is executed on a shard that is allocated on the same server).

The options are to execute the operation on a different thread, or to execute it on the calling thread (note that the API is still async). By default, operationThreaded is set to true which means the operation is executed on a different thread. Here is an example that sets it to false:

DeleteResponse response = client.prepareDelete("twitter", "tweet", "1")

.setOperationThreaded(false)

.get();

## Delete By Query API[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/delete.asciidoc)

The delete by query API allows one to delete a given set of documents based on the result of a query:

BulkByScrollResponse response =

DeleteByQueryAction.INSTANCE.newRequestBuilder(client)

.filter(QueryBuilders.matchQuery("gender", "male")) https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

.source("persons") https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png

.get(); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png

long deleted = response.getDeleted(); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-delete-by-query.html#CO1-1) | query |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-delete-by-query.html#CO1-2) | index |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-delete-by-query.html#CO1-3) | execute the operation |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-delete-by-query.html#CO1-4) | number of deleted documents |

As it can be a long running operation, if you wish to do it asynchronously, you can call execute instead of get and provide a listener（由于执行过程可能 很漫长，可以使用execute来使得操作是异步的，并设置一个监听器来处理执行结果） like:

DeleteByQueryAction.INSTANCE.newRequestBuilder(client)

.filter(QueryBuilders.matchQuery("gender", "male")) https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

.source("persons") https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png

.execute(new ActionListener<BulkByScrollResponse>() { https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png

@Override

public void onResponse(BulkByScrollResponse response) {

long deleted = response.getDeleted(); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png

}

@Override

public void onFailure(Exception e) {

// Handle the exception

}

});

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-delete-by-query.html#CO2-1) | query |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-delete-by-query.html#CO2-2) | index |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-delete-by-query.html#CO2-3) | listener |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-delete-by-query.html#CO2-4) | number of deleted documents |

## Update API[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/update.asciidoc)

You can either create an UpdateRequest and send it to the client:

UpdateRequest updateRequest = new UpdateRequest();

updateRequest.index("index");

updateRequest.type("type");

updateRequest.id("1");

updateRequest.doc(jsonBuilder()

.startObject()

.field("gender", "male")

.endObject());

client.update(updateRequest).get();

Or you can use prepareUpdate() method:

client.prepareUpdate("ttl", "doc", "1")

.setScript(new Script("ctx.\_source.gender = \"male\"" https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png , ScriptService.ScriptType.INLINE, null, null))

.get();

client.prepareUpdate("ttl", "doc", "1")

.setDoc(jsonBuilder() https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png

.startObject()

.field("gender", "male")

.endObject())

.get();

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-update.html#CO3-1) | Your script. It could also be a locally stored script name（可以调用本地存储的脚本）. In that case, you’ll need to use ScriptService.ScriptType.FILE |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-update.html#CO3-2) | Document which will be merged（合并即更新，而不是直接替换） to the existing one. |

Note that you can’t provide both script and doc（不能同时提供script和doc）.

### Update by script[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/update.asciidoc)

The update API allows to update a document based on a script provided:

UpdateRequest updateRequest = new UpdateRequest("ttl", "doc", "1")

.script(new Script("ctx.\_source.gender = \"male\""));

client.update(updateRequest).get();

### Update by merging documents[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/update.asciidoc)

The update API also support passing a partial document, which will be merged into the existing document (simple recursive merge, inner merging of objects, replacing core "keys/values" and arrays). For example:

UpdateRequest updateRequest = new UpdateRequest("index", "type", "1")

.doc(jsonBuilder()

.startObject()

.field("gender", "male")

.endObject());

client.update(updateRequest).get();

### Upsert[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/update.asciidoc)

There is also support for upsert. If the document does not exist, the content of the upsert element will be used to index the fresh doc:

IndexRequest indexRequest = new IndexRequest("index", "type", "1")

.source(jsonBuilder()

.startObject()

.field("name", "Joe Smith")

.field("gender", "male")

.endObject());

UpdateRequest updateRequest = new UpdateRequest("index", "type", "1")

.doc(jsonBuilder()

.startObject()

.field("gender", "male")

.endObject())

.upsert(indexRequest); 创建索引时也可以使用IndexRequest https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

client.update(updateRequest).get();

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-update.html#CO4-1) | If the document does not exist, the one in indexRequest will be added |

If the document index/type/1 already exists, we will have after this operation a document like:

{

"name" : "Joe Dalton",

"gender": "male" https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

}

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-update.html#CO5-1) | This field is added by the update request |

If it does not exist, we will have a new document:

{

"name" : "Joe Smith",

"gender": "male"

}

## Multi Get API[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/multi-get.asciidoc)

The multi get API allows to get a list of documents based on their index, type and id:

MultiGetResponse multiGetItemResponses = client.prepareMultiGet()

.add("twitter", "tweet", "1") https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

.add("twitter", "tweet", "2", "3", "4") https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png

.add("another", "type", "foo") https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png

.get();

for (MultiGetItemResponse itemResponse : multiGetItemResponses) { https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png

GetResponse response = itemResponse.getResponse();

if (response.isExists()) { https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/5.png

String json = response.getSourceAsString(); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/6.png

}

}

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-multi-get.html#CO6-1) | get by a single id |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-multi-get.html#CO6-2) | or by a list of ids for the same index / type |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-multi-get.html#CO6-3) | you can also get from another index |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-multi-get.html#CO6-4) | iterate over the result set |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/5.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-multi-get.html#CO6-5) | you can check if the document exists |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/6.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-multi-get.html#CO6-6) | access to the \_source field |

For more information on the multi get operation, check out the REST [multi get](https://www.elastic.co/guide/en/elasticsearch/reference/5.6/docs-multi-get.html) docs.

## Bulk API[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/bulk.asciidoc)

The bulk API allows one to index and delete several documents in a single request. Here is a sample usage:

import static org.elasticsearch.common.xcontent.XContentFactory.\*;

BulkRequestBuilder bulkRequest = client.prepareBulk();

// either use client#prepare, or use Requests# to directly build index/delete requests

bulkRequest.add(client.prepareIndex("twitter", "tweet", "1")

.setSource(jsonBuilder()

.startObject()

.field("user", "kimchy")

.field("postDate", new Date())

.field("message", "trying out Elasticsearch")

.endObject()

)

);

bulkRequest.add(client.prepareIndex("twitter", "tweet", "2")

.setSource(jsonBuilder()

.startObject()

.field("user", "kimchy")

.field("postDate", new Date())

.field("message", "another post")

.endObject()

)

);

BulkResponse bulkResponse = bulkRequest.get();

if (bulkResponse.hasFailures()) {

// process failures by iterating through each bulk response item

}

## Using Bulk Processor[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/bulk.asciidoc)

The BulkProcessor class offers a simple interface to flush bulk operations automatically based on the number or size of requests(自动或者隔一段时间根据请数量处理操作), or after a given period.

To use it, first create a BulkProcessor instance:

import org.elasticsearch.action.bulk.BackoffPolicy;

import org.elasticsearch.action.bulk.BulkProcessor;

import org.elasticsearch.common.unit.ByteSizeUnit;

import org.elasticsearch.common.unit.ByteSizeValue;

import org.elasticsearch.common.unit.TimeValue;

BulkProcessor bulkProcessor = BulkProcessor.builder(

client, https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

new BulkProcessor.Listener() {

@Override

public void beforeBulk(long executionId,

BulkRequest request) { ... } https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png

@Override

public void afterBulk(long executionId,

BulkRequest request,

BulkResponse response) { ... } https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png

@Override

public void afterBulk(long executionId,

BulkRequest request,

Throwable failure) { ... } https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png

})

.setBulkActions(10000) https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/5.png

.setBulkSize(new ByteSizeValue(5, ByteSizeUnit.MB)) https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/6.png

.setFlushInterval(TimeValue.timeValueSeconds(5)) https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/7.png

.setConcurrentRequests(1) https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/8.png

.setBackoffPolicy(

BackoffPolicy.exponentialBackoff(TimeValue.timeValueMillis(100), 3)) https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/9.png

.build();

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-bulk-processor.html#CO7-1) | Add your elasticsearch client |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-bulk-processor.html#CO7-2) | This method is called just before bulk is executed. You can for example see the numberOfActions with request.numberOfActions() |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-bulk-processor.html#CO7-3) | This method is called after bulk execution. You can for example check if there was some failing requests with response.hasFailures() |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-bulk-processor.html#CO7-4) | This method is called when the bulk failed and raised a Throwable |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/5.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-bulk-processor.html#CO7-5) | We want to execute the bulk every 10 000 requests |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/6.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-bulk-processor.html#CO7-6) | We want to flush the bulk every 5mb |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/7.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-bulk-processor.html#CO7-7) | We want to flush the bulk every 5 seconds whatever the number of requests |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/8.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-bulk-processor.html#CO7-8) | Set the number of concurrent requests. A value of 0 means that only a single request will be allowed to be executed. A value of 1 means 1 concurrent request is allowed to be executed while accumulating new bulk requests. |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/9.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-docs-bulk-processor.html#CO7-9) | Set a custom backoff policy which will initially wait for 100ms, increase exponentially and retries up to three times. A retry is attempted whenever one or more bulk item requests have failed with an EsRejectedExecutionException which indicates that there were too little compute resources available for processing the request. To disable backoff, pass BackoffPolicy.noBackoff(). |

By default, BulkProcessor:

* sets bulkActions to 1000
* sets bulkSize to 5mb
* does not set flushInterval
* sets concurrentRequests to 1, which means an asynchronous execution of the flush operation.
* sets backoffPolicy to an exponential backoff with 8 retries and a start delay of 50ms. The total wait time is roughly 5.1 seconds.

### Add requests[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/bulk.asciidoc)

Then you can simply add your requests to the BulkProcessor:

bulkProcessor.add(new IndexRequest("twitter", "tweet", "1").source(/\* your doc here \*/));

bulkProcessor.add(new DeleteRequest("twitter", "tweet", "2"));

### Closing the Bulk Processor[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/bulk.asciidoc)

When all documents are loaded to the BulkProcessor it can be closed by using awaitClose or closemethods:

bulkProcessor.awaitClose(10, TimeUnit.MINUTES);

or

bulkProcessor.close();

Both methods flush any remaining documents and disable all other scheduled flushes if they were scheduled by setting flushInterval. If concurrent requests were enabled the awaitClose method waits for up to the specified timeout for all bulk requests to complete then returns true, if the specified waiting time elapses before all bulk requests complete, false is returned. The closemethod doesn’t wait for any remaining bulk requests to complete and exits immediately.

### Using Bulk Processor in tests[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/docs/bulk.asciidoc)

If you are running tests with elasticsearch and are using the BulkProcessor to populate your dataset you should better set the number of concurrent requests to 0 so the flush operation of the bulk will be executed in a synchronous manner:

BulkProcessor bulkProcessor = BulkProcessor.builder(client, new BulkProcessor.Listener() { /\* Listener methods \*/ })

.setBulkActions(10000)

.setConcurrentRequests(0)

.build();

// Add your requests

bulkProcessor.add(/\* Your requests \*/);

// Flush any remaining requests

bulkProcessor.flush();

// Or close the bulkProcessor if you don't need it anymore

bulkProcessor.close();

// Refresh your indices

client.admin().indices().prepareRefresh().get();

// Now you can start searching!

client.prepareSearch().get();

[Java API Administration](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin.html)

* [**Indices Administration**](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html)
* [Cluster Administration](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html)

## Indices Administration[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/admin/indices/index.asciidoc)

To access indices Java API, you need to call indices() method from an [AdminClient](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin.html" \o "Java API Administration)（使用adminClient才能访问索引api）:

IndicesAdminClient indicesAdminClient = client.admin().indices();

Note

In the rest of this guide, we will use client.admin().indices().

### Create Index[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/admin/indices/create-index.asciidoc)

Using an [IndicesAdminClient](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html" \o "Indices Administration), you can create an index with all default settings and no mapping:

client.admin().indices().prepareCreate("twitter").get();

##### Index Settings[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/admin/indices/create-index.asciidoc)

Each index created can have specific settings associated with it.

client.admin().indices().prepareCreate("twitter")

.setSettings(Settings.builder() https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

.put("index.number\_of\_shards", 3)

.put("index.number\_of\_replicas", 2)

)

.get(); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO54-1) | Settings for this index |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO54-2) | Execute the action and wait for the result |

### Put Mapping[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/admin/indices/put-mapping.asciidoc)

The PUT mapping API allows you to add a new type while creating an index:

client.admin().indices().prepareCreate("twitter") https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

.addMapping("tweet", "{\n" + https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png

" \"tweet\": {\n" +

" \"properties\": {\n" +

" \"message\": {\n" +

" \"type\": \"string\"\n" +

" }\n" +

" }\n" +

" }\n" +

" }")

.get();

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO55-1) | [Creates an index](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#java-admin-indices-create-index) called twitter |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO55-2) | It also adds a tweet mapping type. |

The PUT mapping API also allows to add a new type to an existing index:

client.admin().indices().preparePutMapping("twitter") https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

.setType("user") https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png

.setSource("{\n" + https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png

" \"properties\": {\n" +

" \"name\": {\n" +

" \"type\": \"string\"\n" +

" }\n" +

" }\n" +

"}")

.get();

// You can also provide the type in the source document

client.admin().indices().preparePutMapping("twitter")

.setType("user")

.setSource("{\n" +

" \"user\":{\n" + https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png

" \"properties\": {\n" +

" \"name\": {\n" +

" \"type\": \"string\"\n" +

" }\n" +

" }\n" +

" }\n" +

"}")

.get();

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO56-1) | Puts a mapping on existing index called twitter |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO56-2) | Adds a user mapping type. |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO56-3) | This user has a predefined type |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO56-4) | type can be also provided within the source |

You can use the same API to update an existing mapping:

client.admin().indices().preparePutMapping("twitter") https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

.setType("user") https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png

.setSource("{\n" + https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png

" \"properties\": {\n" +

" \"user\_name\": {\n" +

" \"type\": \"string\"\n" +

" }\n" +

" }\n" +

"}")

.get();

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO57-1) | Puts a mapping on existing index called twitter |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO57-2) | Updates the user mapping type. |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO57-3) | This user has now a new field user\_name |

### Refresh[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/admin/indices/refresh.asciidoc)

The refresh API allows to explicitly refresh one or more index:

client.admin().indices().prepareRefresh().get(); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

client.admin().indices()

.prepareRefresh("twitter") https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png

.get();

client.admin().indices()

.prepareRefresh("twitter", "company") https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png

.get();

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO58-1) | Refresh all indices |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO58-2) | Refresh one index |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO58-3) | Refresh many indices |

### Get Settings[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/admin/indices/get-settings.asciidoc)

The get settings API allows to retrieve settings of index/indices:

GetSettingsResponse response = client.admin().indices()

.prepareGetSettings("company", "employee").get(); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

for (ObjectObjectCursor<String, Settings> cursor : response.getIndexToSettings()) { https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png

String index = cursor.key; https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png

Settings settings = cursor.value; https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png

Integer shards = settings.getAsInt("index.number\_of\_shards", null); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/5.png

Integer replicas = settings.getAsInt("index.number\_of\_replicas", null); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/6.png

}

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO59-1) | Get settings for indices company and employee |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO59-2) | Iterate over results |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO59-3) | Index name |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO59-4) | Settings for the given index |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/5.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO59-5) | Number of shards for this index |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/6.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO59-6) | Number of replicas for this index |

### Update Indices Settings[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/admin/indices/update-settings.asciidoc)

You can change index settings by calling:

client.admin().indices().prepareUpdateSettings("twitter") https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

.setSettings(Settings.builder() https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png

.put("index.number\_of\_replicas", 0)

)

.get();

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO60-1) | Index to update |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-indices.html#CO60-2) | Settings |

## Cluster Administration[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/admin/cluster/index.asciidoc)

To access cluster Java API, you need to call cluster() method from an [AdminClient](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin.html" \o "Java API Administration):

ClusterAdminClient clusterAdminClient = client.admin().cluster();

Note

In the rest of this guide, we will use client.admin().cluster().

### Cluster Health[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/admin/cluster/health.asciidoc)

#### Health[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/admin/cluster/health.asciidoc)

The cluster health API allows to get a very simple status on the health of the cluster and also can give you some technical information about the cluster status per index:

ClusterHealthResponse healths = client.admin().cluster().prepareHealth().get(); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

String clusterName = healths.getClusterName(); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png

int numberOfDataNodes = healths.getNumberOfDataNodes(); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png

int numberOfNodes = healths.getNumberOfNodes(); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png

for (ClusterIndexHealth health : healths.getIndices().values()) { https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/5.png

String index = health.getIndex(); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/6.png

int numberOfShards = health.getNumberOfShards(); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/7.png

int numberOfReplicas = health.getNumberOfReplicas(); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/8.png

ClusterHealthStatus status = health.getStatus(); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/9.png

}

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO61-1) | Get information for all indices |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO61-2) | Access the cluster name |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO61-3) | Get the total number of data nodes |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO61-4) | Get the total number of nodes |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/5.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO61-5) | Iterate over all indices |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/6.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO61-6) | Index name |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/7.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO61-7) | Number of shards |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/8.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO61-8) | Number of replicas |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/9.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO61-9) | Index status |

#### Wait for status[edit](https://github.com/elastic/elasticsearch/edit/5.6/docs/java-api/admin/cluster/health.asciidoc)

You can use the cluster health API to wait for a specific status for the whole cluster or for a given index:

client.admin().cluster().prepareHealth() https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

.setWaitForYellowStatus() https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png

.get();

client.admin().cluster().prepareHealth("company") https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png

.setWaitForGreenStatus() https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png

.get();

client.admin().cluster().prepareHealth("employee") https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/5.png

.setWaitForGreenStatus() https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/6.png

.setTimeout(TimeValue.timeValueSeconds(2)) https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/7.png

.get();

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO62-1) | Prepare a health request |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO62-2) | Wait for the cluster being yellow |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/3.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO62-3) | Prepare the health request for index company |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/4.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO62-4) | Wait for the index being green |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/5.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO62-5) | Prepare the health request for index employee |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/6.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO62-6) | Wait for the index being green |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/7.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO62-7) | Wait at most for 2 seconds |

If the index does not have the expected status and you want to fail in that case, you need to explicitly interpret the result:

ClusterHealthResponse response = client.admin().cluster().prepareHealth("company")

.setWaitForGreenStatus() https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png

.get();

ClusterHealthStatus status = response.getIndices().get("company").getStatus();

if (!status.equals(ClusterHealthStatus.GREEN)) {

throw new RuntimeException("Index is in " + status + " state"); https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png

}

|  |  |
| --- | --- |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/1.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO63-1) | Wait for the index being green |
| [https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/images/icons/callouts/2.png](https://www.elastic.co/guide/en/elasticsearch/client/java-api/5.6/java-admin-cluster.html#CO63-2) | Throw an exception if not GREEN |