

Elasticsearch Reference Guide Refresh

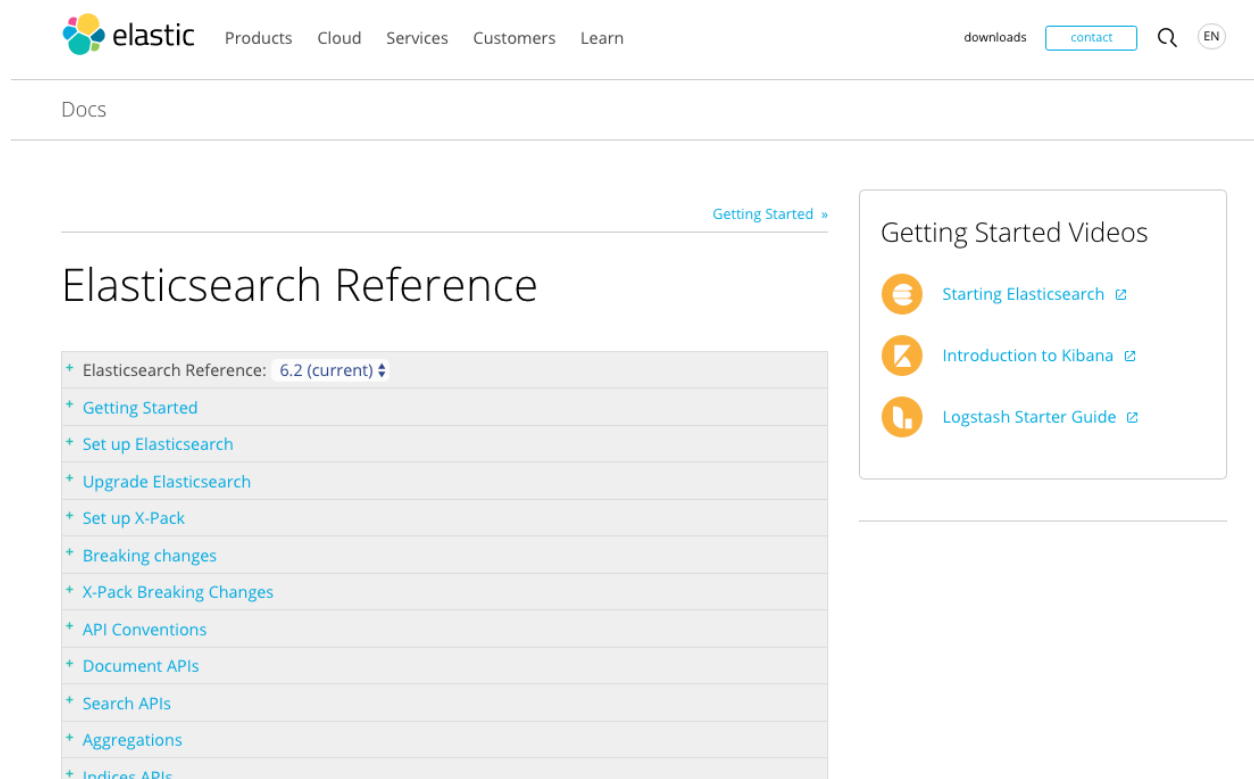
Purpose

The Elasticsearch Reference Guide is difficult to use as it is currently written and structured. The purpose of this document is to illustrate how changes in document structure, API reference content and searchability would improve the usability of the document for end users, particularly for administrators and application developers.

Screen captures of key pages for the discussion are included inline (all captured 5/11/2018).

Basic Guide Structure

The [current guide](#) is accessible in HTML form from the [elastic](#) web site.



The screenshot shows the Elasticsearch Reference Guide website. The header includes the Elastic logo and navigation links: Products, Cloud, Services, Customers, Learn, downloads, contact, and a search icon. The main content area is titled 'Elasticsearch Reference' and features a 'Getting Started' link. A dropdown menu is open, showing a list of topics: Elasticsearch Reference: 6.2 (current), Getting Started, Set up Elasticsearch, Upgrade Elasticsearch, Set up X-Pack, Breaking changes, X-Pack Breaking Changes, API Conventions, Document APIs, Search APIs, Aggregations, and Indices APIs. On the right, there is a 'Getting Started Videos' section with three video thumbnails: 'Starting Elasticsearch', 'Introduction to Kibana', and 'Logstash Starter Guide'.

Structural Problems

The most critical issue with the content is that the guide does not clearly separate important conceptual and performance information about how the Elasticsearch product works from API reference. This makes it difficult to locate the right API endpoint and figure out how to use it

given a specific task, and also makes it difficult to find important conceptual and performance information that should be factored in at design and configuration time.

Another problem is that documentation for X-Pack (a set of extensions that must be separately installed) is unevenly intermingled with core product documentation. Some sections (e.g. Set up X-Pack) are obviously specific to X-Pack. Other sections (e.g. Monitoring Elasticsearch, not shown in the screen capture) require X-Pack but are not clearly labeled in the table of contents.

Revised Structure

Many weaknesses in this guide could be addressed by an overall restructuring and cleanup of existing content.

As a primary organizing principle, sections of content should align with the needs of typical engineering roles in a large-scale development project, grouping information that is needed at different stages of software design and development. The guide should separate concerns to provide a conceptual and functional overview (architects, developers), installation and administration procedures (operations, devops), API reference (developers, devops), and performance guidelines.

In addition, since the X-Pack extension is installed and versioned separately, and because X-Pack APIs are grouped separately and have different conventions, it would be cleaner to present a clear division between X-Pack and core product.

Applying these principles, the content could be roughly regrouped and interlinked as follows.

- Getting Started
 - What is Elasticsearch? [Brief intro paragraph and link to *About Elasticsearch*]
 - Install Elasticsearch [link to *Installing and Configuring Elasticsearch - Installation*]
 - Explore your cluster
 - Create and index data
 - Modify your data
 - Explore your data
 - Learn about Elasticsearch APIs [link to *Elasticsearch API reference*]
- About Elasticsearch
 - Basic Concepts
 - Reading and Writing Documents
 - Mapping
 - Indexing
 - Using an Ingest Node
 - Search
 - Query DSL
 - Aggregations
 - Analysis
 - Testing

- Glossary of Terms
- Elasticsearch API Reference
 - API Overview (brief summary of the scope of the APIs)
 - API Conventions
 - Document APIs
 - Ingest APIs
 - Search APIs
 - cat APIs
 - Cluster APIs
 - Indices APIs
- Installing and Configuring Elasticsearch
 - Installation
 - Installing and using X-Pack [short description and link to *X-Pack*]
 - Upgrade
 - Breaking Changes
 - Elasticsearch configuration
 - Modules
 - Index Modules
 - System configuration
 - Bootstrap Checks
 - Stopping Elasticsearch
 - Release Notes
- Performance
 - Working with result sets
 - Indexing
 - Tuning for indexing speed
 - Tuning for search speed
 - Tuning for disk usage
 - Advanced recipes
- X-Pack
 - Installing X-Pack
 - Configuring Monitoring
 - Configuring Security
 - Configuring X-Pack Java Clients
 - X-Pack Settings
 - Bootstrap Checks
 - X-Pack APIs
 - X-Pack Commands
 - X-Pack Release Notes
 - X-Pack Breaking Changes

API Reference

As noted above, a primary usability problem with this guide is that it does not clearly separate important conceptual and performance information about how the Elasticsearch product works from API reference. Conceptual information embedded in the API documents should be removed from API reference and integrated into the “About Elasticsearch” portion of the guide with appropriate links to and from the API reference.

Another issue specific to the API documentation is inconsistent use of terminology. The term “API” seems to be used both to talk about functional groupings of endpoints (e.g. Document APIs) and to talk about functional subsets, individual endpoints or even individual methods on endpoints within those groupings (e.g. Get API). Adhering to consistent terminology, particularly if it can be realigned to industry standard REST language, would make it easier for developers to understand the APIs.

Finally, poor and confusing coverage of shared conventions that are used throughout the APIs (which should also be referenced from individual endpoint documentation) further reduces usability of the material.

API Endpoints

A documentation template with predictable structure and content for every endpoint would make the API reference sections much easier for developers to understand at a glance. This approach would also significantly reduce the wordiness of the document and make it easier to read and maintain. Ideally, this structure could be developed and managed as an OpenAPI specification and exposed via Swagger UI or similar tooling; unfortunately given the divergence of the design of the APIs from standard REST conventions OpenAPI is not a good fit.

The current reference guide groups API endpoints into several functional groups (e.g. Document APIs, Search APIs). Ideally, the introductory page for each functional group should provide a concise overview of the scope of the functionality, with an easy way to navigate to the structured detail on how to use specific endpoints and methods. The Document APIs page provides a good example of how to modify the content to improve usability for developers.

Document APIs - Current Introduction

The introductory page for the Document APIs is primarily a group of links that separates operations on single-documents from operations on multiple documents. There are several key issues to address in order to improve the documentation.

Most critically, the introductory page does not provide a concise overview of the functional scope of the Document APIs. It assumes that the user will understand why the data replication model matters as applied to the API, and it assumes that the user understands the CRUD acronym.

Finally, the major organization of links on the introductory page into the content should be mirrored by the links in the right-hand sidebar.

Docs

Document APIs

This section starts with a short introduction to Elasticsearch's [data replication model](#), followed by a detailed description of the following CRUD APIs:

Single document APIs

- [Index API](#)
- [Get API](#)
- [Delete API](#)
- [Update API](#)

Multi-document APIs

- [Multi Get API](#)
- [Bulk API](#)
- [Delete By Query API](#)
- [Update By Query API](#)
- [Reindex API](#)

NOTE All CRUD APIs are single-index APIs. The `index` parameter accepts a single index name, or an `alias` which points to a single index.

[URL-based access control](#) [Reading and Writing documents](#)

- Starting Elasticsearch
- Introduction to Kibana
- Logstash Starter Guide

- Elasticsearch Reference: 6.2 (current)
- Getting Started
- Set up Elasticsearch
- Upgrade Elasticsearch
- Set up X-Pack
- Breaking changes
- X-Pack Breaking Changes
- API Conventions
- Document APIs**
 - Reading and Writing documents
 - Index API
 - Get API
 - Delete API
 - Delete By Query API
 - Update API
 - Update By Query API

Document APIs - Revised Introduction

Applying these concepts, the introduction could be rewritten to be clearly task-based, with each bullet point consisting of a link to the documentation of the path, method, parameters and examples. It could look as follows, with right-hand navigation links reordered and retitled to match.

The Document APIs are a set of endpoints that expose CRUD (create, read, update, delete) and statistical operations on documents indexed in Elasticsearch. Some endpoints operate on a single document at a time; others are bulk or query-based endpoints that can operate over multiple documents.

Single Document Operations

- [Read a document \(GET\)](#)
- [Read a document's term vectors \(GET\)](#)
- [Create or update a document in the index \(PUT, POST\)](#)
- [Update a document partially or with a script \(POST\)](#)
- [Delete a document \(DELETE\)](#)

Multiple Document Operations

- Read documents (GET)
- Read term vectors for multiple documents (POST)
- Copy documents to a new index (POST)
- Create, update or delete documents in bulk (POST)
- Update documents by query (POST)
- Delete documents by query (POST)

Note: To better design robust workflows and error handling with these APIs, read more about the underlying Elasticsearch *data replication model* [link].

API Conventions — Content

The *API Conventions* section says that it documents common parameters and other conventions that are used throughout the APIs. It does not, however, document basic conventions that are normally shared across endpoints of a REST API, including a discussion of patterns in path construction, security, authorization and authentication, supported methods, response codes, and any headers that are used consistently throughout. These omissions are a significant usability problem for developers who expect to find this information in a central place.

The content that it does cover is uneven (some of the parameters covered are not common across all APIs), and are presented as wordy text, often without examples. A better alternative is to restrict this section to parameters that are universal across all endpoints, presenting them in a format that consistently shows parameter names, types, default values and specific usage notes.

API Conventions — Enhanced Content

Applying these concepts, the API conventions section should consist of the following:

- Path structure and components (new content)
- Methods (new content)
- Security and Access Control (new content)
- Common Response Status Codes (new content)
- Common Parameters (restructured from existing content)
- Specifying Units and Date Math

API Conventions — Common Options

The common options section is particularly difficult to use. It mixes documentation of parameters with documentation of how to format parameter values, in no discernable order.

Elasticsearch Reference [6.2] » API Conventions » Common options

[« Date math support in index names](#) [URL-based access control »](#)

Common options

The following options can be applied to all of the REST APIs.

Pretty Results

When appending `?pretty=true` to any request made, the JSON returned will be pretty formatted (use it for debugging only!). Another option is to set `?format=yaml` which will cause the result to be returned in the (sometimes) more readable yaml format.

Human readable output

Statistics are returned in a format suitable for humans (eg `"exists_time": "1h"` or `"size": "1kb"`) and for computers (eg `"exists_time_in_millis": 3600000` or `"size_in_bytes": 1024`). The human readable values can be turned off by adding `?human=false` to the query string. This makes sense when the stats results are being consumed by a monitoring tool, rather than intended for human consumption. The default for the `human` flag is `false`.

Date Math

Most parameters which accept a formatted date value — such as `gt` and `lt` in [range queries](#) range queries, or `from` and `to` in [daterange aggregations](#) — understand date maths.

Getting Started Videos

- [Starting Elasticsearch](#)
- [Introduction to Kibana](#)
- [Logstash Starter Guide](#)

On this page

- [Pretty Results](#)
- [Human readable output](#)
- [Date Math](#)
- [Response Filtering](#)
- [Flat Settings](#)
- [Parameters](#)
- [Boolean Values](#)
- [Number Values](#)
- [Time units](#)

Specification of how to construct date math, boolean and number values and other units should be separated into its own section. Documentation of common parameters should be structured rather than narrative, and should always include examples. For example:

Common Query Path Parameters

pretty	
Return pretty-printed JSON response. Not recommended for production, enable for debugging only.	Type: boolean Required? no Default: false

(Example request and response here, with curl copy and console controls as in the rest of the document...)

human	
Enhance statistics output format to include human-readable values in addition to default machine-readable values.	Type: boolean Required? no Default: false

(Example request and response here, with curl copy and console controls as in the rest of the document...)

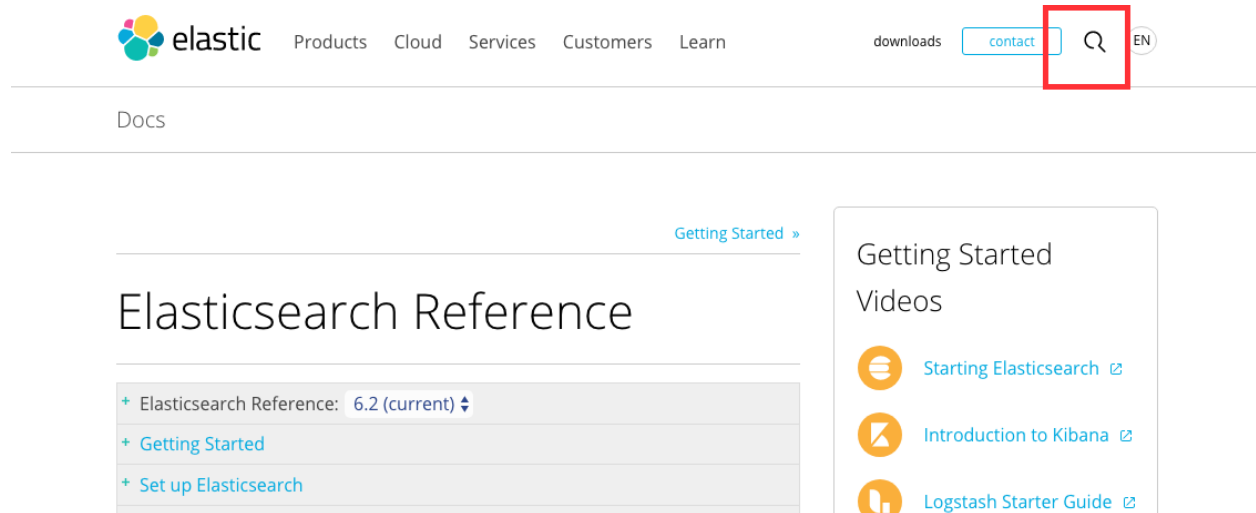
Common Request Headers

Content-Type	
Specifies the type of content contained in a request body.	Type: string Required? yes Default: N/A Supported values (e.g. "application/json") vary by endpoint; endpoint documentation includes a list of supported content types.

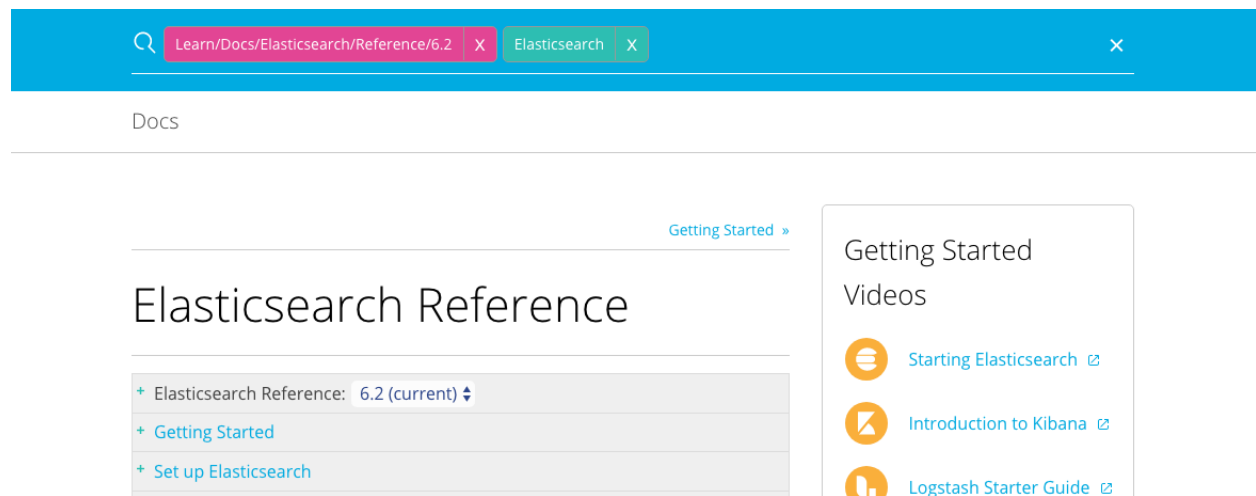
(Example request and response here, with curl copy and console controls as in the rest of the document...)

Searchability of the Guide

The product documentation is searchable using the elastic site search tool (annotated in the screen capture below).



When accessed from the reference guide pages, search is constrained to the context of the reference guide.



Constrained search over the guide is a powerful feature of the documentation and useful to the developer, but not intuitively presented. An icon to launch constrained search over the reference guide could be added in the visual context of the reference guide (mock-up below). This approach could be user-tested to validate whether introducing a small redundancy in the interface helps users quickly find what they need in the guide.

