Executing CRUD Operations Using the Elasticsearch APIs

Janani Ravi
CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Operations on indices and indexed documents

- create
- read/retrieve
- update
- delete

Bulk operations on indexed documents

Bulk creation of indices from JSON data

RESTful APIs with Elasticsearch

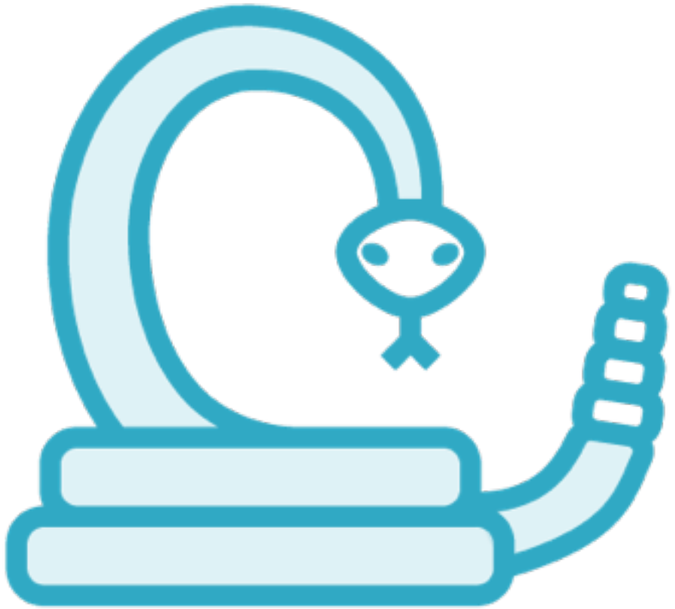
RESTful APIs

Elasticsearch uses REST APIs to administer the cluster, perform CRUD operations, search etc.

RESTful APIs

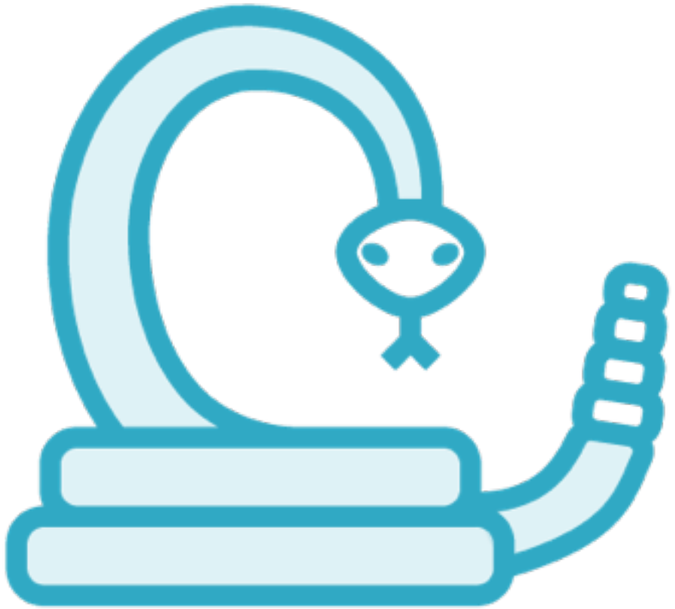
Data is sent to and received from the server in JSON form

cURL for Requests to REST APIs



cURL is a tool which allows you to transfer data from and to a server using a variety of protocols

cURL for Requests to REST APIs



HTTP, FTP, GOPHER, IMAP, LDAP etc.

Demo

Use cURL to make requests and receive responses from servers

Use cURL to list all indices in the cluster

curl -XGET ‘localhost:9200/_cat/indices?v&pretty

To list all the nodes in our current cluster:
curl -XGET ‘localhost:9200/_cat/nodes?v&pretty’

To monitor the health of our current cluster:
curl -XGET ‘localhost:9200/_cat/health?v&pretty’

Demo

Create indices in Elasticsearch

Understand the differences between POST and PUT requests

To create a new index called products: curl -XPUT ‘localhost:9100/products?&pretty’

PUT:
Used to create and update resources on the server. The PUT request is idempotent. It can be run multiple times and reproduce the same result.

POST:
Used to update resources.
It is not idempotent. You can get a different result based on the initial condiitons.

RE Health: index will be yellow if there isn’t a replica assigned in the single-node cluster

Demo

Index individual documents in Elasticsearch

The contents of the document has to be specified in JSON notation within the request body of the HTTP PUT request.

```
->
-->curl -XPUT 'localhost:9200/products/mobiles/1?pretty' -d'
> {
>   "name": "iPhone 7",
>   "camera": "12MP",
>   "storage": "256GB",
>   "display": "4.7inch",
>   "battery": "1,960mAh",
>   "reviews": ["Incredibly happy after having used it for one week", "Best iPhone
( to Android"]
> }
> '
```

Option to specify the unique ID

JUST REALISED SHE HAS ALL THE COMMANDS IN A TEXT FILE :')

If you don't specify an id, elasticsearch will automatically assign one

Demo

Retrieve documents from Elasticsearch by id:

- whole documents

- partial documents

“found”: false means that the document doesn’t exist

We can reduce the data transfer that is made across the network by not retrieving the source contents of the document (_source=false)

We can also use_source to specify specific fields we want to retrieve

Demo

To update the whole document: Use the same URL as document creation, but a pre-existing ID. This indicates to ES that the operation is an update and not a create.

Update by using a PUT request and changing the body

You can see if a document has been updated by looking at the _Version

Update documents by id:

- whole documents
- partial documents

To make partial updates to a document: use _update API, which is invoked by using a POST command with a doc field in the request body

If you update a document with exactly the same fields as the latest version of the document, the results will be equal to “noop” and the version will remain the same.

We can use the _update API and a script to programmatically manipulate data within a document

Demo

Delete documents in an index

Delete the entire index

When you delete a document from elasticsearch it isn't actually deleted but is marked for deletion. Later on, when a merge happens behind the scenes in order to consolidate space, that is when the actual deletion takes place.

Check if a particular document exists within an index and document type by using HEAD HTTP request.

Use -XDELETE flag

To delete the entire index, use the delete request with only the name of the index.

Demo

The _mget command allows you to retrieve multiple documents with one command. Specify the documents needed in the request body.

You can specify both the index and document type in the URL (if they are the same) and then you only need to specify the ID within the request body.

Bulk operations on documents:

- retrieve multiple documents

- index multiple documents

- multiple operations in one command

The _bulk command allows you to specify multiple operations in one request. e.g. indexing multiple documents in one go.

If any of the operations in the bulk result in an error, ES will still process the other operations.

To autogenerate the ID field just leave it out of the request body.

If you specify new ids when updating, the original documents will remain untouched and new documents will be created.

In addition to the index keyword, you can also use the create keyword in order to add a new document to the index.

Demo

Bulk index documents from a JSON file

The “index” field is required. You can choose to add attributes to this such as id, type, index or leave it blank.

The -H argument along with cURL allows you to specify that you want to read a file of content type JSON. A POST request is used because you want to update the index

You do not have to create the index nor the document type upfront. Adding documents to an index results in the creation of that index.

Summary

Performed CRUD operations on indexes holding documents

Implemented bulk operations on indexed documents

Created indices in bulk from JSON data in a file