Elastic Search Cheat Sheet

Basic Manipulations

Delete index

```
curl -X DELETE 'http://<ip_add:port>/<index>'
```

Create index

```
curl -X PUT "http://<ip\_add:port>/<index>?pretty"
-H 'Content-Type: application/json' -d'
{
  "settings": {
    "number_of_shards": ns,
    "number_of_replicas": nr
  }
}
'
```

List all indices

```
curl -X GET 'http://<ip_add:port>/_cat/indices?v'
```

List all docs in index

```
curl -X GET 'http://<ip_add:port>/<index>/_search'
```

Add a document to an index

Index API Adds a JSON document to the specified data stream or index and makes it searchable. If the target is an index and the document already exists, the request updates the document and increments its version.

Automatically create data streams and indices

If no mapping exists, the index operation creates a dynamic mapping.

```
curl -X PUT "<ip_add:port>/_cluster/settings?pretty"
-H 'Content-Type: application/json' -d'
{
  "persistent": {
    "action.auto_create_index": "true"
  }
}
'
```

If no id is provided, the id will be generated automatically.

```
curl -X PUT "http://<ip_add:port>/<index>/_doc/<id_doc>
?pretty" -H 'Content-Type: application/json' -d'
{
  "key0":"value0"
  }'
```

retrieve a document

```
curl -X GET "http://<ip_add:port>/<index>/_doc/<id_doc>
?pretty"
```

Update a document

Elasticsearch documents are immutable, Update API retrieve the concerned document, change it fields values and index it.

Update part of a document

```
curl -X POST "http://<ip_add:port>/<index>/_update/
<id_doc>?pretty" -H 'Content-Type: application/json' -d'
{
  "doc": {
    "key0": "value1"
    or ( add attribute)
    "key1": "value2"
  }
}'
```

Scripted Updates

```
curl -X POST "http://<ip_add:port>/<index>/_update/
<id_doc>?pretty" -H 'Content-Type: application/json' -d'
{
  "script" : {
    "source": "ctx._source.<field1> += params.parm1",
    "params" : {
      "parm1" : "value1"
    }
  }
}
'
```

By default updates that don't change anything detect that they don't change anything and return **"result": "noop"**

Ex: if field1's value is different from params.parm then increment it by 1 else skip. Why not only use as condition `ctx._source.<field1> != params.parm1` ?
As discussed at each update of doc it is removed and replaced with another one, using this condition even if it is respected the doc will be removed and replaced with another one with the same content.

```
curl -X POST "http://<ip_add:port>/<index>/_update/
<id_doc>?pretty" -H 'Content-Type: application/json' -d'
{
  "script" : {
    "source": """
        if (ctx._source.<field1> == params.parm1){
            ctx.op='noop';
        }
        """
        ctx._source.<field1> ++ ;
    "params" : {
      "parm1" : "value1"
    }
  }
}'
```

Upserts

If the document does not already exist, the contents of the upsert element are inserted as a new document. If the document exists, the script is executed:

```
curl -X POST "localhost:9200/test/_update/1?pretty"
-H 'Content-Type: application/json' -d'
{
  "script": {
    "source": "ctx._source.counter += params.count",
    "params": {
      "count": 4
    }
  },
  "upsert": {
    "counter": 1
  }
}'
```