**BULK Insert**

example

POST /bookdb\_index/book/\_bulk

{ "index": { "\_id": 1 }}

{ "title": "Elasticsearch: The Definitive Guide", "authors": ["clinton gormley", "zachary tong"], "summary" : "A distibuted real-time search and analytics engine", "publish\_date" : "2015-02-07", "num\_reviews": 20, "publisher": "oreilly" }

{ "index": { "\_id": 2 }}

{ "title": "Taming Text: How to Find, Organize, and Manipulate It", "authors": ["grant ingersoll", "thomas morton", "drew farris"], "summary" : "organize text using approaches such as full-text search, proper name recognition, clustering, tagging, information extraction, and summarization", "publish\_date" : "2013-01-24", "num\_reviews": 12, "publisher": "manning" }

{ "index": { "\_id": 3 }}

{ "title": "Elasticsearch in Action", "authors": ["radu gheorge", "matthew lee hinman", "roy russo"], "summary" : "build scalable search applications using Elasticsearch without having to do complex low-level programming or understand advanced data science algorithms", "publish\_date" : "2015-12-03", "num\_reviews": 18, "publisher": "manning" }

{ "index": { "\_id": 4 }}

{ "title": "Solr in Action", "authors": ["trey grainger", "timothy potter"], "summary" : "Comprehensive guide to implementing a scalable search engine using Apache Solr", "publish\_date" : "2014-04-05", "num\_reviews": 23, "publisher": "manning" }

**MATCH**

GET /index name/type/\_search?q=key:value

GET /index name/type/\_search?q=value

(or)

POST /index name/type/\_search

{

query:{

match:{

"key":"value"

}

}

}

query:{

match:{

"query":"value"

}

}

**Multi-MATCH**

query:{

multi\_match:{

"query":"value",

"fields":["field1,"field2",....]

}

}

**Display selected fields**

query:{

match:{

"key":"value"

}

"\_source":["field1","field2",.....]

"highlight":{

"fields":{

"fieldname":{}

}

}

}

**Boosting in certain fields**

query:{

multi\_match:{

"query":"value",

"fields":["field1,"field2^3",....] //^3 increases the importance of field2 by factor 3

}

}

**Fuzzy Query**

query:{

multi\_match:{

"key":"value" // value might be misspelled

"fields":["field1,"field2",....]

"fuzziness":"AUTO"

}

}

**Wildcard**

query:{

wildcard: {

"key":"value" // value can contain \* or ? for ex)"author":"t\*" //author name starts with t. ? matches any character and \* matches zero or more characters

}

}

**REGEXP**

query:{

regexp: {

"key":"value" // value can contain regular expression ex)t[a-z]1

}

}

**Match phrase**

query:{

multi\_match:{

"query":"word1 word2",

"fields":["field1,"field2",....],

"type":"phrase", // all the words specified must be in specified order in any one of the specified fields.

"slop":"3" //the number of other words inbetween the specified words can be 3.

}

}

**Query String**

// It can contain bools, regexp, fuzziness, wildcard, range etc

query:{

"query\_string":{

"query":"(value1) AND (value2) OR (value3)",

"fields":["field1,"field2^3",....]

},

"\_source":[],

"highlight":{

"fields":{

"field1":{},

}

}

}

**Match exactly what we want ()term/terms**

query:{

term:{

"key":"value" ex)"author":"abc"

}

"\_source":["field1","field2",.....]

"highlight":{

"fields":{

"fieldname":{} ex)"author":{}

}

}

}

**term - sorted**

query:{

term:{

"key":["value1","value2","value3"...] ex)"author":"abc"

}

"\_source":["field1","field2",.....]

"highlight":{

"fields":{

"fieldname":{} ex)"author":{}

}

}

sort:[

{"field name":{"order":"desc/asc"}}

]

}

**Range queries**

query:{

range:{

"field name":{

"gte":"",

"lte":"",

"gt":"",

"lt":""

}

}

"\_source":["field1","field2",.....]

}

**Filtered bool query**

query:{

"filtered":{

"query":{

"match":{

"key":"value"

}

}

}

"filter":{

"range":{

"field1":{

"gte":20

}

}

}

}

query:{

"filtered":{

"query":{

"match":{

"key":"value"

}

}

}

"filter":{

"bool":{

"must":{ //AND

"range":{

"field1":{

"gte":20

}

}

},

"must\_not":{ //NOT

"range":{

"field1":{

"lte":10

}

}

},

"should":{ //OR

"term":{

"key":"value"

}

}

}

}

}