CREATE INDEX

```
PUT /products
{
 "settings": {
   "number_of_shards": 2,
   "number_of_replicas": 2
 }
}
```

DELETE AN INDEX

```
DELETE /pages/
```

ADD DOCUMENT TO AN INDEX

```
POST /products/_doc
{
 "name": "Coffee Maker",
 "price": 64,
 "in_stock": 10
}
```

ADD DOCUMENT WITH SPECIFIC ID TO AN INDEX

```
PUT /products/_doc/100
{
 "name": "Toaster",
 "price": 49,
 "in_stock": 4
}
```

RETTRIEVE DATA

```
GET /products/_doc/100
```

UPDATE DATA

```
POST /products/_update/100
{
 "doc": {
   "in_stock": 3
 }
}
```

ADD NEW FIELD

```
POST /products/_update/100
{
 "doc": {
   "tags": ["electronics"]
 }
}
```

WRITE SCRIPT TO REDUCE THE IN_STOCK

```
POST /products/_update/100
{
 "script": {
   "source": "ctx._source.in_stock--"
```

```
  }
}
```

SCRIPT WITH PARAMETERS

```
POST /products/_update/100
{
  "script": {
    "source": "ctx._source.in_stock -= params.quantity",
    "params": {
      "quantity": 4
    }
  }
}
```

GET AND UPDATE OR CREATE

```
POST /products/_update/101
{
  "script": {
    "source": "ctx._source.in_stock++"
  },
  "upsert": {
    "name": "Blender",
    "price": 399,
    "in_stock": 5
  }
}
```

REPLACE DOCUMENT

```
PUT /products/_doc/100
{
  "name": "Toaster",
  "price": 79,
  "in_stock": 4
}
```

DELETE A DOCUMENT

```
DELETE /products/_doc/101
```

PRIMARY_TERM AND SEQUENCE_NO

```
POST /products/_update/100?if_primary_term=1&if_seq_no=10
{
  "doc": {
    "in_stock": 123
  }
}
```

UPDATE BY QUERY

```
POST /products/_update_by_query
{
  "conflicts": "proceed",
  "script": {
    "source": "ctx._source.in_stock--"
  },
  "query": {
    "match_all": {

    }
  }
}
```

SEARCH

```
GET /products/_search
{
  "query": {
    "match_all": {}
  }
}
```

DELETE BY QUERY

```
POST /products/_delete_by_query
{
  "query": {
    "match_all": {}
  }
}
```

BULK QUERY

```
POST /_bulk
{"index": {"_index":"products", "_id": 200}}
{"name": "Espresso Machine", "price":199, "in_stock": 5}
{"create": {"_index": "products", "_id": 201}}
{"name": "Milk frother", "price":149, "in_stock": 240}
```

```
POST /products/_bulk
{"update": {"_id": 201}}
{"doc": {"price":129}}
{"delete": {"_id": 200}}
```

ANALYZER

```
POST /_analyze
{
  "text": "2 guys walk into  a bar, but the third... DUCKS! :-)",
  "analyzer": "standard"
}
```

```
POST /_analyze
{
  "text": "2 guys walk into  a bar, but the third... DUCKS! :-)",
  "char_filter": [],
  "tokenizer": "standard",
  "filter": ["lowercase"]
}
```

MAPPING

```
PUT /reviews
{
  "mappings": {
    "properties": {
      "rating": {"type": "float"},
      "content": {"type": "text"},
      "product_id": {"type": "integer"},
      "author": {
        "properties": {
          "first_name": {"type": "text"},
          "last_name": {"type": "text"},
          "email": {"type": "keyword"}
        }
      }
    }
  }
}
```

```
PUT /reviews/_mapping
{
  "properties": {
    "created_at": {"type": "date"}
  }
}
```

```
PUT /reviews/_doc/1
{
  "rating": 5.0,
  "content": "Outstanding course!! Bo really",
  "product_id": 123,
  "author": {
    "first_name": "John",
    "last_name": "Doe",
    "email": "johndoe123@gmail.com"
  }
}
```

```
PUT /reviews/_doc/5
{
  "rating": 4.5,
  "content": "Very useful",
  "product_id": 123,
  "created_at": "1436011284000",
  "author": {
    "first_name": "Taylor",
    "last_name": "West",
    "email": "twest@example.com"
  }
}
```

```
PUT /reviews/_mapping
{
  "properties": {
    "author": {
      "properties": {
        "email": {
          "type": "keyword",
          "ignore_above": 256
        }
      }
    }
  }
}
```

```
POST /_reindex
{
  "source": {
    "index": "reviews"
  }
  , "dest": {
    "index": "reviews_new"
  }
}
```

```
PUT /reviews/_mapping
{
  "properties": {
    "comment": {
      "type": "alias",
      "path": "content"
    }
  }
}
```

```
GET /reviews/_search
{
  "query": {
    "match": {
      "comment": "outstanding"
    }
  }
}
```

RETRIEVE MAPPING

```
GET /reviews/_mapping
GET /reviews/_mapping/field/content
GET /reviews/_mapping/field/author.email
```

MULTI FIELD

```
PUT /multi_field_test
{
  "mappings": {
    "properties": {
      "description": {
        "type": "text"
      },
      "ingredients": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword"
          }
        }
      }
    }
  }
}


POST /multi_field_test/_doc
{
  "description": "To make this Spaghetti, you first need to...",
  "ingredients": ["Spaghetti", "Becon", "Eggs"]
}
```

TEMPLATE

```
PUT /_template/access-logs
{
  "index_patterns": ["access-logs-*"],
  "settings": {
    "number_of_shards": 2,
    "index.mapping.coerce": false
  },
  "mappings": {
    "properties": {
      "@timestamp": {
        "type": "date"
      },
      "url.original": {
        "type": "keyword"
      },
      "http.request.referrer": {
        "type": "keyword"
      },
      "http.response.status_code": {
        "type": "long"
      }
    }
  }
}

PUT /access-logs-2020-01-01
```

CONTROL DYNAMIC MAPPING

```
PUT /people
{
  "mappings": {
    "dynamic": "strict",
    "properties": {
      "first_name": {
        "type": "text"
      }
    }
  }
}
```

```
curl -H "Content-Type: application/x-ndjson" -XPOST http://localhost:9200/products/_bulk --data-binary “@products-bulk.json"
```

SEARCH QUERY

```
GET /products/_search?q=*
GET /products/_search?q=name:lobster
GET /products/_search?q=tags:Meat AND name:Tuna
GET /products/_search
{
  "query": {
    "match_all": {}
  }
}
GET /products/_search?explain
{
  "query": {
    "term": {
      "name": "lobster"
    }
  }
}
```

TERM SEARCH QUERY(RANGE, DATE, BOOLEAN)

```
GET /products/_search
{
  "query": {
    "term": {
      "is_active": {
        "value": true
      }
    }
  }
}

GET /products/_search
{
  "query": {
    "terms": {
      "tags.keyword": [
        "Soup",
        "Cake"
      ]
    }
  }
}

GET /products/_search
{
  "query": {
    "ids": {
      "values": [1, 2, 3]
    }
  }
}

GET /products/_search
{
  "query": {
    "range": {
      "in_stock": {
        "gte": 1,
        "lte": 5
      }
    }
  }
}

GET /products/_search
{
  "query": {
    "range": {
      "created": {
        "gte": "2010/01/01",
        "lte": "2010/12/31"
      }
    }
  }
}

GET /products/_search
{
  "query": {
    "range": {
      "created": {
        "gte": "01-01-2010",
        "lte": "31-12-2010",
        "format": "dd-MM-yyyy"
      }
    }
  }
}

GET /products/_search
{
  "query": {
    "range": {
      "created": {
        "gte": "2010/01/01||-1y"
      }
    }
  }
}

GET /products/_search
{
  "query": {
    "range": {
      "created": {
        "gte": “2010/01/01||-1y-1d"
      }
    }
  }
```

```
}

GET /products/_search
{
  "query": {
    "range": {
      "created": {
        "gte": "2010/01/01||/M-1y-1d"
      }
    }
  }
}

GET /products/_search
{
  "query": {
    "range": {
      "created": {
        "gte": "now"
      }
    }
  }
}
```

SEARCH WITH EXIST

```
GET /products/_search
{
  "query": {
    "exists": {
      "field": "tags"
    }
  }
}
```

SEARCH WITH PREFIX

```
GET /products/_search
{
  "query": {
    "prefix": {
      "tags.keyword": "Vege"
    }
  }
}
```

SEARCH WITH WILDCARDS

```
GET /products/_search
{
  "query": {
    "wildcard": {
      "tags.keyword": "Veg*ble"
    }
  }
}

GET /products/_search
{
  "query": {
    "wildcard": {
      "tags.keyword": "Veg?ble"
    }
  }
}

GET /products/_search
{
  "query": {
    "wildcard": {
      "tags.keyword": "Veget?ble"
    }
  }
}
```

SEARCH WITH REGULAR EXPRESSION

```
GET /products/_search
{
  "query": {
    "regexp": {
      "tags.keyword": "Veg[a-zA-Z]+ble"
    }
  }
}
```

MATCHING QUERY

```
GET /products/_search
{
  "query": {
    "match": {
      "title": "Recipes with pasta"
    }
  }
}

GET /products/_search
{
  "query": {
    "match": {
      "title": {
        "query": "Recipes with pasta",
        "operator": "and"
      }
    }
  }
}
```

MATCHING PHRASES

```
 GET /products/_search
{
  "query": {
    "match_phrase": {
      "title": "Recipes with pasta"
    }
  }
}
```

MULTIPLE FIELD QUERY

```
GET /products/_search
{
  "query": {
    "multi_match": {
      "query": "vegetable",
      "fields": [
        "name",
        "description"
      ]
    }
  }
}
```

QUERYING WITH BOOLEAN LOGIC

```
GET /products/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "ingradient.name": "parmesan"
          }
        }
      ],
      "must_not": [
        {
          "match": {
            "ingradient.name": "tuna"
          }
        }
      ],
      "should": [
        {
          "match": {
            "ingradient.name": "parsley"
          }
        }
      ],
      "filter": [
        {
          "range": {
            "ingredient.name": {
              "lte": 15
            }
          }
        }
      ]
    }
  }
}
```

DEBUGGING BOOL QUERIES WITH NAME

```
GET /products/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "ingradient.name": {
              "query": "parmesan",
              "_name": "parmesan_must"
            }
          }
        }
      ],
      "must_not": [
        {
          "match": {
            "ingradient.name": {
              "query": "tuna",
              "_name": "tuna_must_not"
            }
          }
        }
      ],
      "should": [
        {
          "match": {
            "ingradient.name": {
              "query": "parsley",
              "_name": "parsley_should"
            }
          }
        }
      ],
      "filter": [
        {
          "range": {
            "ingredient.name": {
              "lte": 15,
              "_name": "prep_time_filter"
            }
          }
        }
      ]
    }
  }
}
```

DEBUG SEARCH QUERY

```
GET /products/default/1/_explain
{
  "query": {
    "term": {
      "name": "lobster"
    }
  }
}
```

QUERYING NESTED OBJECTS

```
PUT /department?include_type_name=true
{
  "mappings": {
    "_doc": {
      "properties": {
        "name": {
          "type": "text"
        },
        "employee": {
          "type": "nested"
        }
      }
    }
  }
}

POST /department/_doc/1
{
  "name": "Development",
  "employee": [
    {
      "name": "Eric Green",
      "age": 39,
      "gender": "M",
      "position": "Big data specialist"
    },
    {
      "name": "James Taylor",
      "age": 27,
      "gender": "M",
      "position": "Software developer"
    },
    {
      "name": "Gary Jenkins",
      "age": 21,
      "gender": "M",
      "position": "Intern"
    },
    {
      "name": "Julie Powell",
      "age": 26,
      "gender": "F",
      "position": "Intern"
    },
    {
      "name": "Benjamin Smith",
      "age": 46,
      "gender": "M",
      "position": "Senior Software engineer"
    }
  ]
}

GET /department/_search
{
  "query": {
    "nested": {
      "path": "employee",
      "query": {
        "bool": {
          "must": [
            {
              "match": {
                "employee.position": "intern"
              }
            },
            {
              "term": {
                "employee.gender.keyword": {
                  "value": "F"
                }
              }
            }
          ]
        }
      }
    }
  }
}
```

INNER HITS

```
GET /department/_search
{
  "_source": false,
  "query": {
    "nested": {
      "path": "employee",
      "inner_hits": {},
      "query": {
        "bool": {
          "must": [
            {
              "match": {
                "employee.position": "intern"
              }
            },
            {
              "term": {
                "employee.gender.keyword": {
                  "value": "F"
                }
              }
            }
          ]
        }
      }
    }
  }
}
```

MAPPING DOCUMENT RELATIONSHIPS

```
PUT /department
{
  "mappings": {
    "properties": {
      "join_field": {
        "type": "join",
        "relations": {
          "department": "employee"
        }
      }
    }
  }
}
```

ADDING DOCUMENTS

```
PUT /department/_doc/1
{
  "name": "Development",
  "join_field": "department"
}

PUT /department/_doc/3?routing=1
{
  "name": "Bo Andersen",
  "age": 28,
  "gender": "M",
  "join_field": {
    "name": "employee",
    "parent": 1
  }
}

PUT /department/_doc/4?routing=2
{
  "name": "John Doe",
  "age": 44,
  "gender": "M",
  "join_field": {
    "name": "employee",
    "parent": 2
  }
}
```

QUERYING BY PARENT ID

```
GET /department/_search
{
  "query": {
    "parent_id": {
      "type": "employee",
      "id": 1
    }
  }
}
```

QUERYING CHILD DOCUMENTS BY PARENT

```
GET /department/_search
{
  "query": {
    "has_parent": {
      "parent_type": "department",
      "score": true,
      "query": {
        "term": {
          "name.keyword": "Development"
        }
      }
    }
  }
}
```

QUERYING PARENT BY CHILD DOCUMENTS

```
GET /department/_search
{
  "query": {
    "has_child": {
      "type": "employee",
      "query": {
        "bool": {
          "must": [
            {
              "range": {
                "age": {
                  "gte": 50
                }
              }
            }
          ],
          "should": [
            {
              "term": {
                "gender.keyword": "M"
              }
            }
          ]
        }
      }
    }
  }
}

GET /department/_search
{
  "query": {
    "has_child": {
      "score_mode": "sum",
      "type": "employee",
      "query": {
        "bool": {
          "must": [
            {
              "range": {
                "age": {
                  "gte": 50
                }
              }
            }
          ],
          "should": [
            {
              "term": {
                "gender.keyword": "M"
              }
            }
          ]
        }
      }
    }
  }
}

GET /department/_search
{
  "query": {
    "has_child": {
      "score_mode": "sum",
      "min_children": 2,
      "max_children": 5,
      "type": "employee",
      "query": {
        "bool": {
          "must": [
            {
              "range": {
                "age": {
                  "gte": 50
                }
              }
            }
          ],
          "should": [
            {
              "term": {
                "gender.keyword": "M"
              }
            }
          ]
        }
      }
    }
  }
}
```

MULTI LEVEL RELATIONS

```
PUT company?include_type_name=true
{
  "mappings": {
    "_doc": {
      "properties": {
        "join_field": {
          "type": "join",
          "relations": {
            "company": ["department", "supplier"],
            "department": "employee"
          }
        }
      }
    }
  }
}

PUT /company/_doc/1
{
  "name": "My company Inc",
  "join_field": "company"
}

PUT /company/_doc/2?routing=1
{
  "name": "Development",
  "join_field": {
    "name": "department",
    "parent": 1
  }
}

PUT /company/_doc/3?routing=1
{
  "name": "Boe Andersen",
  "join_field": {
    "name": "employee",
    "parent": 2
  }
}

GET /company/_search
{
  "query": {
    "has_child": {
      "type": "department",
      "query": {
        "has_child": {
          "type": "employee",
          "query": {
            "term": {
              "name.keyword": {
                "value": "John Doe"
              }
            }
          }
        }
      }
    }
  }
}
```

PARENT/CHILD INNER HITS

```
GET /company/_search
{
  "query": {
    "has_child": {
      "type": "employee",
      "inner_hits": {},
      "query": {
        "bool": {
          "must": [
            {
              "range": {
                "age": {
                  "gte": 50
                }
              }
            }
          ],
          "should": [
            {
              "term": {
                "gender.keyword": "M"
              }
            }
          ]
        }
      }
    }
  }
}

GET /department/_search
{
  "query": {
    "has_parent": {
      "parent_type": "department",
      "inner_hits": {},
      "query": {
        "term": {
          "name.keyword": "Development"
        }
      }
    }
  }
}
```

TERMS LOOKUP MECHANISM

```
PUT /user/_doc/1
{
  "name": "John Roberts",
  "following": [2, 3]
}

PUT /stories/_doc/1
{
  "user": 3,
  "content": "Now look, a penguin"
}

GET /stories/_search
{
  "query": {
    "terms": {
      "user": [1, 2]
    }
  }
}

GET /stories/_search
{
  "query": {
    "terms": {
      "user": {
        "index": "users",
        "type": "_doc",
        "id": 1,
        "path": "following"
      }
    }
  }
}
```

SOURCE FILTERING

```
GET /recipes/_search?format=yaml
{
  "_source": false,
  "query": {
    "match": {
      "title": "pasta"
    }
  }
}
```

```
GET /recipes/_search?format=yaml
{
  "_source": "ingredients.name",
  "query": {
    "match": {
      "title": "pasta"
    }
  }
}
```

```
GET /recipes/_search?format=yaml
{
  "_source": "ingredients*",
  "query": {
    "match": {
      "title": "pasta"
    }
  }
}
```

```
GET /recipes/_search?format=yaml
{
  "_source": ["ingredients*", "servings"],
  "query": {
    "match": {
      "title": "pasta"
    }
  }
}
```

```
GET /recipes/_search?format=yaml
{
  "_source": {
    "includes": "ingredients*",
    "excludes": "ingredients.name"
  },
  "query": {
    "match": {
      "title": "pasta"
    }
  }
}
```

SPECIFYING THE RESULT SIZE

```
GET /recipes/_search?size=2
{
  "_source": false,
  "query": {
    "match": {
      "title": "pasta"
    }
  }
}
```

```
GET /recipes/_search
{
  "size": 2,
  "_source": false,
  "query": {
    "match": {
      "title": "pasta"
    }
  }
}
```

SPECIFYING AN OFFSET

```
GET /recipes/_search
{
  "size": 2,
  "_source": false,
  "from": 0,
  "query": {
    "match": {
      "title": "pasta"
    }
  }
}
```

SORTING RESULTS

```
GET /recipes/_search
{
  "_source": false,
  "query": {
    "match_all": {}
  }
  , "sort":
  [
    "preparation_time_minutes"
  ]
}
```

```
GET /recipes/_search
{
  "_source": "created",
  "query": {
    "match_all": {}
  }
  , "sort":
  [
    {
      "created": "desc"

    }
  ]
}
```

```
GET /recipes/_search
{
  "_source": ["preparation_time_minutes","created"],
  "query": {
    "match_all": {}
  }
  , "sort":
  [
    {"preparation_time_minutes": "asc"},
    {"created": "desc"}
  ]
}
```

SORTING BY MULTI VALUE FIELDS

```
GET /recipes/_search
{
  "_source": "ratings",
  "query": {
    "match_all": {}
  }
  , "sort":
  [
    {
      "ratings": {
        "order": "desc",
        "mode": "avg"
      }

    }
  ]
}
```

FILTERS

```
GET /recipes/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "title": "pasta"
          }
        }
      ],
      "filter": [
        {
          "range": {
            "preparation_time_minutes": {
              "lte": 15
            }
          }
        }
      ]
    }
  }
}
```

METRIC AGGREGATIONS

```
GET /order/_search
{
  "size": 0,
  "aggs": {
    "totl_sales": {
      "sum": {
        "field": "total_amount"
      }
    },
    "avg_sale": {
      "avg": {
        "field": "total_amount"
      }
    },
    "min_sale": {
      "min": {
        "field": "total_amount"
      }
    },
    "max_sale": {
      "max": {
        "field": "total_amount"
      }
    }
  }
}

GET /order/_search
{
  "size": 0,
  "aggs": {
    "total_salesmen": {
      "cordinality": {
        "field": "salesman.id"
      }
    }
  }
}

GET /order/_search
{
  "size": 0,
  "aggs": {
    "values_count": {
      "value_count": "total_amount"
    }
  }
}

GET /order/_search
{
  "size": 0,
  "aggs": {
    "amount_stats": {
      "stats": {
        "field": "total_amount"
      }
    }
  }
}
```

PRPXIMITY SEARCHES

```
GET order/_search
{
  "query": {
    "match_phrase": {
      "title": {
        "query": "spicy sauce",
        "slop": 1
      }
    }
  }
}
```

FUZZY MATCH QUERY

```
GET order/_search
{
  "query": {
    "match": {
      "name": {
        "query": "l0bster",
        "fuzziness": "auto"
      }
    }
  }
}
```

HIGHLIGHTING MATCHES IN FIELDS

```
GET order/_search
{
  "_source": false,
  "query": {
    "match": {
      "description": "Elasticsearch"
    }
  },
  "highlight": {
    "pre_tags": [ "<strong>" ],
    "post_tags": [ "</strong>" ],
    "fields": {
      "description": {}
    }
  }
}
```