./elasticsearch -Ecluster.name=spire\_demo\_cluster-dev -Enode.name=spire\_demo\_node-dev &

GET /\_cat/health?v

GET /\_cat/nodes?v

GET /\_cat/indices?v

PUT /customer?pretty

PUT /customer/external/1?pretty

{

"name": "John Doe"

}

GET /customer/external/1?pretty

DELETE /customer?pretty

PUT /customer/external/1?pretty

{

"name": "John Doe"

}

PUT /customer/external/1?pretty - Replace

{

"name": "Jane Doe"

}

POST /customer/external?pretty - index a document without an explicit ID

{

"name": "Jane Doe"

}

POST /customer/external/1/\_update?pretty - Update

{

"doc": { "name": "Jane Doe" }

}

POST /customer/external/1/\_update?pretty - new field

{

"doc": { "name": "Jane Doe", "age": 20 }

}

POST /customer/external/1/\_update?pretty - script update

{

"script" : "ctx.\_source.age += 5"

}

DELETE /customer/external/2?pretty

POST /customer/external/\_delete\_by\_query

{

"query": {

"match\_all": {}

}

}

POST /bank/account/\_bulk?pretty&refresh --data-binary "@accounts.json"

GET /bank/\_search?q=\*&sort=account\_number:asc&pretty

POST /bank/\_search

{

"query": {

"match\_all": {}

},

"sort": [

{

"account\_number": "asc"

}

]

}

POST /bank/\_search

{

"query": {

"match\_all": {}

},

"size": 1

}

POST /bank/\_search

{

"query": {

"match\_all": {}

},

"sort": [

{

"account\_number": "asc"

}

],

"from": 10,

"size": 5

}

POST /bank/\_search

{

"query": {

"match\_all": {}

},

"sort": {

"balance": {

"order": "desc"

}

}

}

POST /bank/\_search

{

"query": {

"match\_all": {}

},

"\_source": [

"account\_number",

"balance"

]

}

POST /bank/\_search - match query

{

"query": {

"match": {

"account\_number": 20

}

}

}

POST /bank/\_search - returns all accounts containing the term "mill" or "lane" in the address

{

"query": {

"match": {

"address": "mill lane"

}

}

}

POST /bank/\_search - returns all accounts containing the phrase "mill lane" in the address

{

"query": {

"match\_phrase": {

"address": "mill lane"

}

}

}

POST /bank/\_search - returns all accounts containing "mill" or "lane" in the address

{

"query": {

"bool": {

"should": [

{

"match": {

"address": "mill"

}

},

{

"match": {

"address": "lane"

}

}

]

}

}

}

POST /bank/\_search - returns all accounts containing "mill" and "lane" in the address

{

"query": {

"bool": {

"must": [

{

"match": {

"address": "mill"

}

},

{

"match": {

"address": "lane"

}

}

]

}

}

}

POST /bank/\_search - returns all accounts that contain neither "mill" nor "lane" in the address

{

"query": {

"bool": {

"must\_not": [

{

"match": {

"address": "mill"

}

},

{

"match": {

"address": "lane"

}

}

]

}

}

}

POST /bank/\_search - returns all accounts of anybody who is 40 years old but doesn’t live in ID

{

"query": {

"bool": {

"must": [

{

"match": {

"age": "40"

}

}

],

"must\_not": [

{

"match": {

"state": "ID"

}

}

]

}

}

}

POST /bank/\_search - returns all accounts with balances between 20000 and 30000 inclusive

{

"query": {

"bool": {

"must": {

"match\_all": {}

},

"filter": {

"range": {

"balance": {

"gte": 20000,

"lte": 30000

}

}

}

}

}

}

POST /bank/\_search - groups all the accounts by state, and then returns the top 10 (default) states sorted by count descending (default)

{

"size": 0,

"aggs": {

"group\_by\_state": {

"terms": {

"field": "state.keyword"

}

}

}

}

POST /bank/\_search - groups and additionally calculates the average account balance by state

{

"size": 0,

"aggs": {

"group\_by\_state": {

"terms": {

"field": "state.keyword"

},

"aggs": {

"average\_balance": {

"avg": {

"field": "balance"

}

}

}

}

}

}

POST /bank/\_search - groups and additionally sorts on the average balance in descending order

{

"size": 0,

"aggs": {

"group\_by\_state": {

"terms": {

"field": "state.keyword",

"order": {

"average\_balance": "desc"

}

},

"aggs": {

"average\_balance": {

"avg": {

"field": "balance"

}

}

}

}

}

}

POST /bank/\_search - groups by age brackets (ages 20-29, 30-39, and 40-49), then by gender, and then finally get the average account balance, per age bracket, per gender

{

"size": 0,

"aggs": {

"group\_by\_age": {

"range": {

"field": "age",

"ranges": [

{

"from": 20,

"to": 30

},

{

"from": 30,

"to": 40

},

{

"from": 40,

"to": 50

}

]

},

"aggs": {

"group\_by\_gender": {

"terms": {

"field": "gender.keyword"

},

"aggs": {

"average\_balance": {

"avg": {

"field": "balance"

}

}

}

}

}

}

}

}

POST /\_analyze

{

"analyzer": "whitespace",

"text": "The quick brown fox."

}

POST /\_analyze

{

"tokenizer": "standard",

"filter": [

"lowercase",

"asciifolding"

],

"text": "Is this déja vu?"

}

PUT /my\_index

{

"settings": {

"analysis": {

"analyzer": {

"std\_folded": {

"type": "custom",

"tokenizer": "standard",

"filter": [

"lowercase",

"asciifolding"

]

}

}

}

},

"mappings": {

"my\_type": {

"properties": {

"my\_text": {

"type": "text",

"analyzer": "std\_folded"

}

}

}

}

}

POST /my\_index/\_analyze

{

"analyzer": "std\_folded",

"text": "Is this déjà vu?"

}

POST /my\_index/\_analyze

{

"field": "my\_text",

"text": "Is this déjà vu?"

}

POST \_search

{

"\_source": "role.roleName",

"suggest": {

"role-suggest" : {

"prefix" : "prduc",

"completion" : {

"field" : "role.aliases",

"fuzzy" : {

"fuzziness" : 2

}

}

}

}

}