Elastic Search

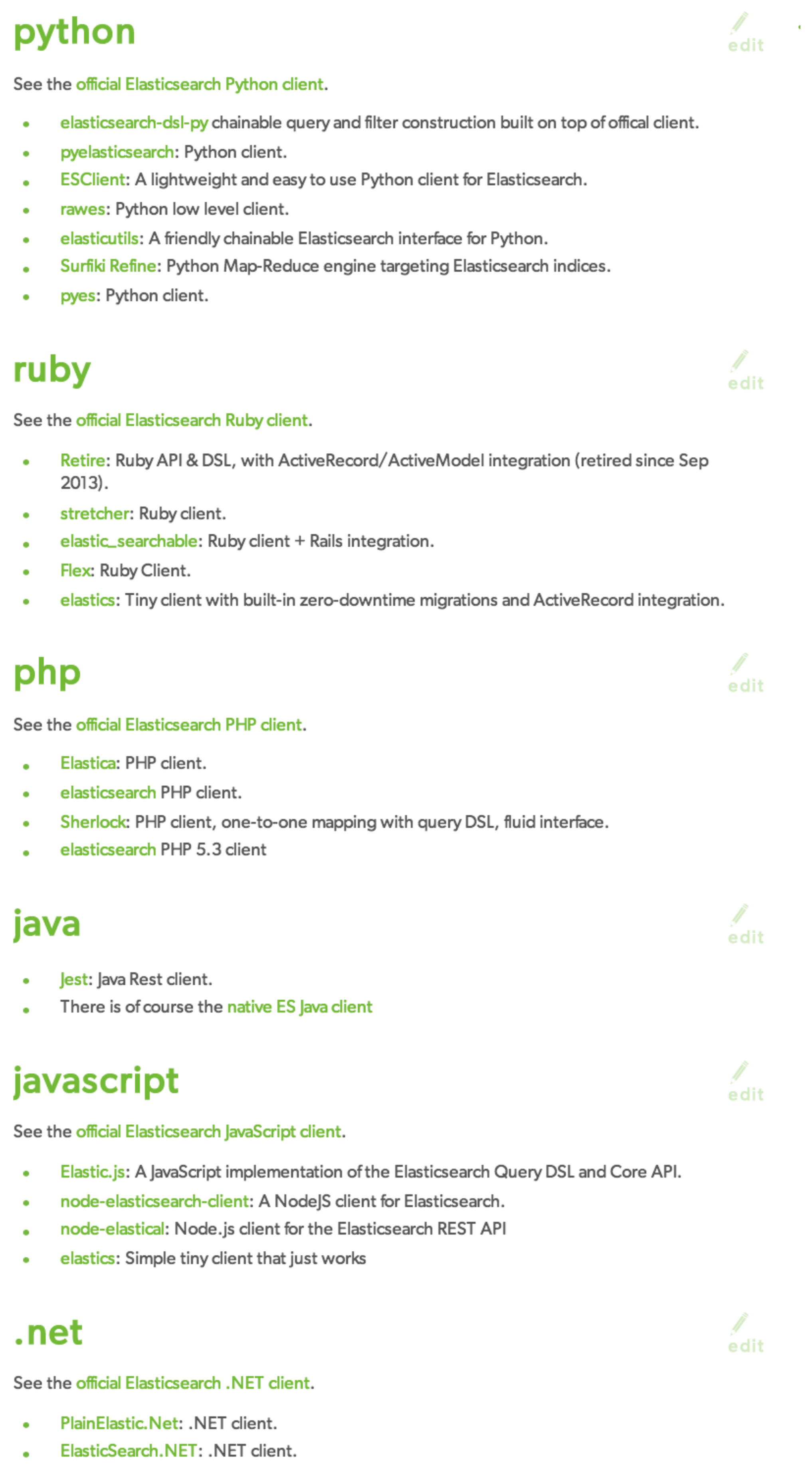
Jakub Čecháček & Andrej Galád

Quick overview

- Fast & Distributed
- Document-Based with JSON
- Schema-less
- Fulltext on top of Apache Lucine
- RESTful interface

APIs

- HTTP RESTful API
- Native Java API
- Client available for many languages.



python

edit

See the official Elasticsearch Python client.

- elasticsearch-dsl-py chainable query and filter construction built on top of offical client.
- pyelasticsearch: Python client.
- ESClient: A lightweight and easy to use Python client for Elasticsearch.
- rawes: Python low level client.
- elasticutils: A friendly chainable Elasticsearch interface for Python.
- Surfiki Refine: Python Map-Reduce engine targeting Elasticsearch indices.
- pyes: Python client.

ruby

edit

See the official Elasticsearch Ruby client.

- Retire: Ruby API & DSL, with ActiveRecord/ActiveModel integration (retired since Sep 2013).
- stretcher: Ruby client.
- elastic_searchable: Ruby client + Rails integration.
- Flex: Ruby Client.
- elastics: Tiny client with built-in zero-downtime migrations and ActiveRecord integration.

php

edit

See the official Elasticsearch PHP client.

- Elastica: PHP client.
- elasticsearch PHP client.
- Sherlock: PHP client, one-to-one mapping with query DSL, fluid interface.
- elasticsearch PHP 5.3 client

java

edit

- Jest: Java Rest client.
- There is of course the native ES Java client

javascript

edit

See the official Elasticsearch JavaScript client.

- Elastic.js: A JavaScript implementation of the Elasticsearch Query DSL and Core API.
- node-elasticsearch-client: A NodeJS client for Elasticsearch.
- node-elastical: Node.js client for the Elasticsearch REST API
- elastics: Simple tiny client that just works

.net

edit

See the official Elasticsearch .NET client.

- PlainElastic.Net: .NET client.
- ElasticSearch.NET: .NET client.

Distributed

- Multiple nodes running in single cluster
- Data are split into shards (# configurable)
    - Zero or more replicas (guaranteed to be on different node)
- Self-managing cluster
    - Automatic master detection (including failover)

Installation

- Requires Java
- Download from http://elasticsearch.org
- Extract the archive
- Run `$ELASTIC_HOME/bin/elasticsearch`
- Notice the name of started node.

How do we use it?

- We will see on next few slides
- You can also try it yourself

  - http://54.93.34.39/

Logical Structure

Relational Systems	Elastic Search
Database	Index
Table	Type
Row	Document
Column	Field

Index documents

- Use HTTP PUT method to store a new document

```
curl -XPUT localhost:9200/dba/question/42 -d
'{ "Title": "How to index a document." }'
```

- Use HTTP POST method to store a new version of document

```
curl -XPOST localhost:9200/dba/question/42 -d
'{ "Title": "How to change a document." }'
```

Get & Delete documents

- Use HTTP GET method to store a new document

```
curl -XGET localhost:9200/dba/question/42
```

- Use HTTP DELETE method to delte a document

```
curl -XDELETE localhost:9200/dba/question/42
```

Search the data

- Query-String searching

```
curl -XGET localhost:9200/dba/question/_search
?q=title:elasticsearch
```

- More powerful search DSL

```
curl -XGET localhost:9200/dba/question/_search -d
'{
  "query": {
    "query_string": {
      "query": "nosql OR title:elasticsearch"
    }
  }
}'
```

Queries

- How well does a document match specified criteria
- **match**
  - Query specified field for a string match
- **multi_match**
  - Query multiple fields for the same match
- **match_phrase**
  - Query for an exact phase
- **match_all**
  - Match all documents

Filters

- **Yes** or **No** question on the fields
- **term**
  - Does a field exactly match given term?
- **range**
  - Is number in specified range?
- **exists** / **missing**
  - Is there a non-null field with specified name?
- Much more is available (see the Filter DSL docs)

Filters + Queries

“Search for all questions about NoSQL asked this year.”

```
curl -XGET localhost:9200/dba/question/_search -d
'{
  "query": {
    "filtered": {
      "query": {               Match NoSQL related
        "multi_match": {
          "query": "NoSQL databases",
          "fields": ["tags^10", "title^5", "_all"]
        }
      },
      "filter": {              Filter 1 year old
        "range": {
          "creation_date": {
            "gt" : "now-1y"
          }
        }
      }
    }
  }
}'
```

```
{
  "took": 88,            ←―――――――――――――――――  Execution time
  "timed_out": false,
  "_shards": {
     "total": 5,
     "successful": 5,
     "failed": 0
  },
  "hits": {              ←―――――――――――――――――  Information about the search
     "total": 893,       ←―――――――――――――――  Number of matched documents
     "max_score": 2.4688244,   ←―――――――――  Rating of document with best match
     "hits": [
        {
           "_index": "dba",          ←―――  Where is the document stored
           "_type": “question",      ←――  What is the type of matched doc
           "_id": “59043",
           "_score": 2.4688244,      ←――  Relevance score of this document
           "_source": {         ←―――――――  The document itself
              "author": {
                 "name": "Lucas Kauffman",
                 "id": 5030
              },
              "rating": 0,
              "body": "...",
              "tags": [
                 "nosql"
              ],
              "comments": [],
              "title": "Elasticsearch: Versioning a document on revisions"
           }
        },
...
}
```

Aggregations

- Collecting analytic information about your data
- Metrics
  - Compute metrics over sets of documents
  - What is the average rating of questions about NoSQL?
- Bucketing
  - Aggregates documents into buckets
  - How many question are there for each tag?

Aggregations (example)

```
curl -XGET localhost:9200/dba/question/_search -d
{
  "fields": ["aggregations"],
  "aggs": {
    "distribution": {
      "terms": {
        "field": "tags",
        "size": 4
      }
    }
  }
}
```

```
"aggregations": {
    "distribution": {
        "doc_count_error_upper_bound": 537,
        "sum_other_doc_count": 56869,
        "buckets": [
            {
                "key": "sql",
                "doc_count": 12388
            },
            {
                "key": "server",
                "doc_count": 10277
            },
            {
                "key": "mysql",
                "doc_count": 7029
            },
            {
                "key": "2008",
                "doc_count": 4142
            }
        ]
    }
}
```

Relationships

ElasticSearch provides 2 types of mechanisms

- **Nested Documents**
  - Index time join
  - Efficiently stored in Lucine
  - Use case: “Comments” on “Post”
- **Paren / Child documents**
  - Query time join
  - Links documents based on parent / child id
  - One-to-Many / Many-to-One relation
  - User case: “Answers” to “Question”

Schema-less

- ES will dynamically index any new field
- Type of the field will be guessed
- Often we know our data, at least partially
- Can we use this knowledge?

Mapping

- Define how ES searches our data
- Completely optional
- Data must be re-indexed after mapping change

Mapping (continued)

- Analysers (stop words, language, not analysed)
- Field types
- Specify document relationships

```
curl -XGET localhost:9200/dba/answer/_mapping
```

```
"answer": {
   "_parent": { "type": "question" }, ←        Parent document type
    "properties": {  ←                          Field mappings
       "accepted": { "type": "boolean" },
       "author": {
          "properties": {
             "id": { "type": "long" },
             "name": { "type": "string" }
          }
       },
       "body": { "type": "string" },
       "comments": {
          "type": "nested", ←                   Index as nested documents
          "properties": {
             "author": { … },
             "body": { "type": "string" },
             "creation_date": {
                "type": "date",
                "format": "dateOptionalTime"
             },
             "rating": { "type": "long" }
          }
       },
       "creation_date": { … },
       "rating": { "type": "long"} ←            This field is of type long
    }
  }
}
```

Any questions?