SEACOM
the leading open source architects

PyCon Nove

elastic

Python and Elasticsearch: from Text Search to NLP and beyond

Dario Balinzo

Pycon9 - Firenze, 22/04/2018

whoami

Software Engineer @ Seacom

dariob@seacom.it

What is Elasticsearch?

Rapid Query Execution

Sophisticated Query Language

API First Engine

Advance Controls

Horizontal Scale

Rich Out-of-the-box Frameworks

Python Clients

elasticsearch-py

- official low level client
- pip install elasticsearch

elasticsearch-dsl

- High level client
- pip install elasticsearch-dsl

Elastic: a few concepts

Elastic: a few concepts

SQL	Elasticsearch
-Database	-Index
-Table	-Type
-Record	-Document


```
twitter/post/15565
{
    "user" : "kimchy",
    "post_date" : 2009-11-15T14:12:12",
    "message" : "trying out Elasticsearch"
}
```

Elastic: a few concepts

```
"mappings": {
  "post": {
        "properties": {
            "user":     { "type": "keyword"},
            "post_date":{ "type": "date" },
            "message": { "type": "text" }
        }
  }
}
```

Let’s code!

elasticsearch-py: Indexing

```
from elasticsearch import Elasticsearch
es = Elasticsearch()

request_body = {
"mappings": { "post": {"properties": {
   "user":     { "type": "keyword"   },
   "post_date":  { "type": "date"      },
   "message":    { "type": "text"      }
                              }
                 }
            }
   }

res = es.indices.create(index="twitter",
      body=request_body)
```

elasticsearch-py: Indexing

```
from datetime import datetime
from elasticsearch import Elasticsearch
es = Elasticsearch()

doc = {
    'user': 'kimchy',
    'post_date': datetime.now(),
    'message': 'Elasticsearch: cool. bonsai cool.',
    }

res = es.index(index="twitter", doc_type='post',
id=15565, body=doc)
print(res['created'])
```

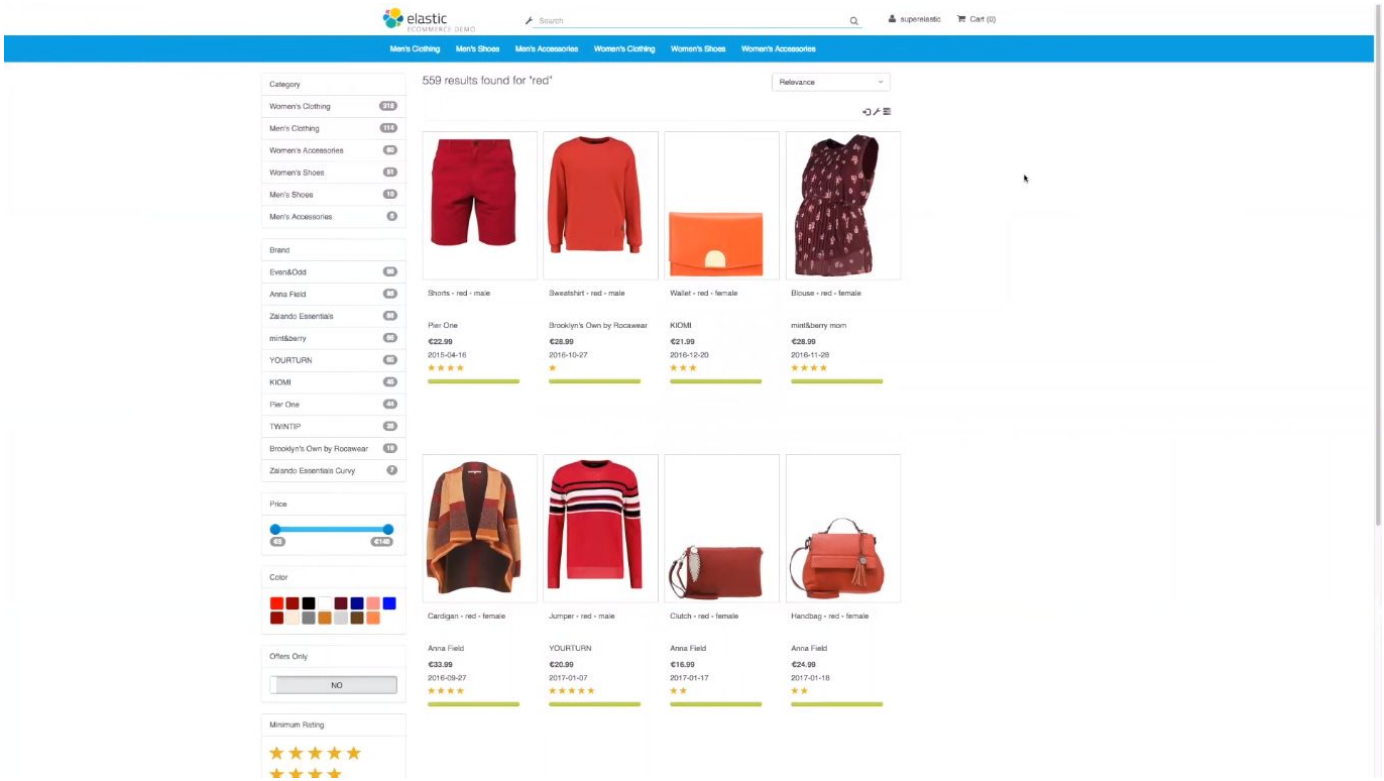
elasticsearch-dsl: defining a metamodel

```
class Comment(DocType):
    author = Keyword()
    message = Text(analyzer='english')
    post_date = Date()

  class Meta:
        index = 'twitter'

  def save(self, ** kwargs):
        self.post_date = datetime.now()
        return super().save(** kwargs)
```

elasticsearch-dsl: indexing

```
# create the mappings in Elasticsearch
  Post.init()
# instantiate the document
first = Post(
    user='kimchy',
    message='Elasticsearch: cool.bonsai
         cool.')
  # every document has an id in meta
  first.meta.id = 16658
  # save the document into the cluster
  first.save()
  # fetch a document
  doc = Post.get(id=16658)
```

Searching


elastic
Search
superelastic
Cart (0)
Men's Clothing
Men's Shoes
Men's Accessories
Women's Clothing
Women's Shoes
Women's Accessories
Category
Women's Clothing
Men's Clothing
Women's Accessories
Women's Shoes
Men's Shoes
Men's Accessories
Brand
Even&Odd
Anna Field
Zalando Essentials
mint&berry
YOURTURN
KIOMI
Pier One
TWINTIP
Brooklyn's Own by Rocawear
Zalando Essentials Curvy
Price
Color
Offers Only
NO
Minimum Rating
559 results found for "red"
Relevance
Shorts - red - male
Pier One
€22.99
2015-04-16
Sweatshirt - red - male
Brooklyn's Own by Rocawear
€28.99
2016-10-27
Wallet - red - female
KIOMI
€21.99
2016-12-20
Blouse - red - female
mint&berry mom
€28.99
2016-11-28
Cardigan - red - female
Anna Field
€33.99
2016-09-27
Jumper - red - male
YOURTURN
€20.99
2017-01-07
Clutch - red - female
Anna Field
€16.99
2017-01-17
Handbag - red - female
Anna Field
€24.99
2017-01-18

Inverted Index

1: Winter is coming.
2: Ours is the fury.
3: The choice is yours.

term	freq	documents
choice	1	3
coming	1	1
fury	1	2
is	3	1, 2, 3
ours	1	2
the	2	2, 3
winter	1	1
yours	1	3

Dictionary

Postings

Text Search

```
# match: search on title only
 s = Search(index="my-index").query("match",
 title="the python book")
 response = s.execute()

# multi match: search on title and body
q = Q("multi_match", query='python django',
fields=['title', 'body'])
s = s.query(q)
response = s.execute()
```

Queries combination

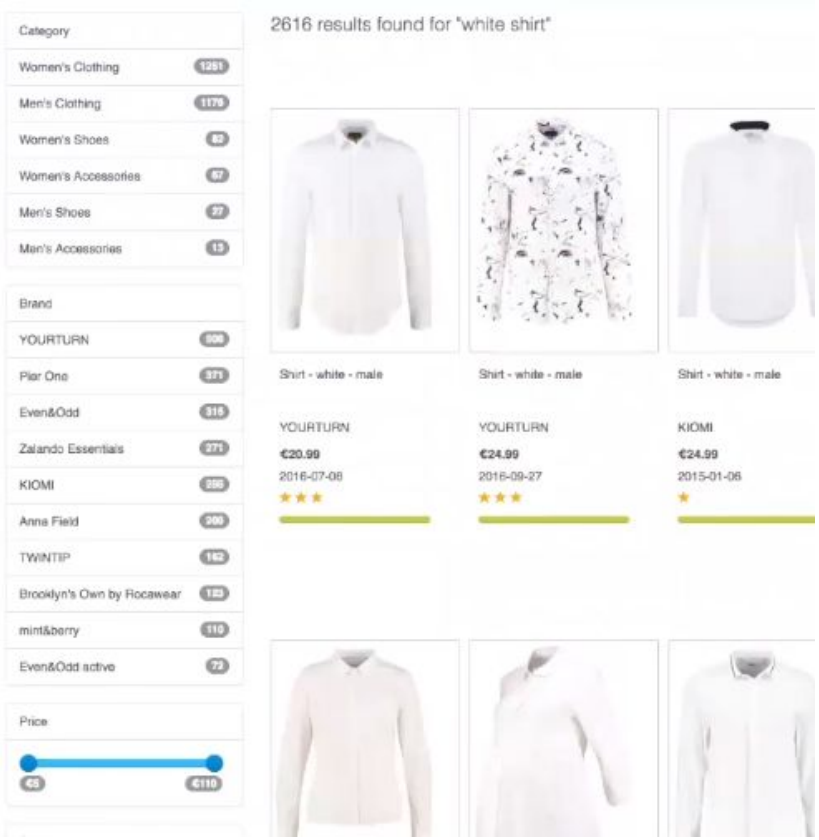
```
Q("match", title='python') | Q("match",
title='django')
# {"bool": {"should": [...]}}

Q("match", title='python') & Q("match",
title='django')
# {"bool": {"must": [...]}}

~Q("match", title="python")
# {"bool": {"must_not": [...]}}
```

Custom results order

```
# sorting
s = Search().sort(
    'category',
    '-title',
    {"lines" : {"order" : "desc"}}
)

# Pagination
s = s[10:20]
# {"from": 10, "size": 10}

# Rescoring
s.extra(rescore={'window_size': 50, "query":
{"rescore_query": Q(...)}})
```

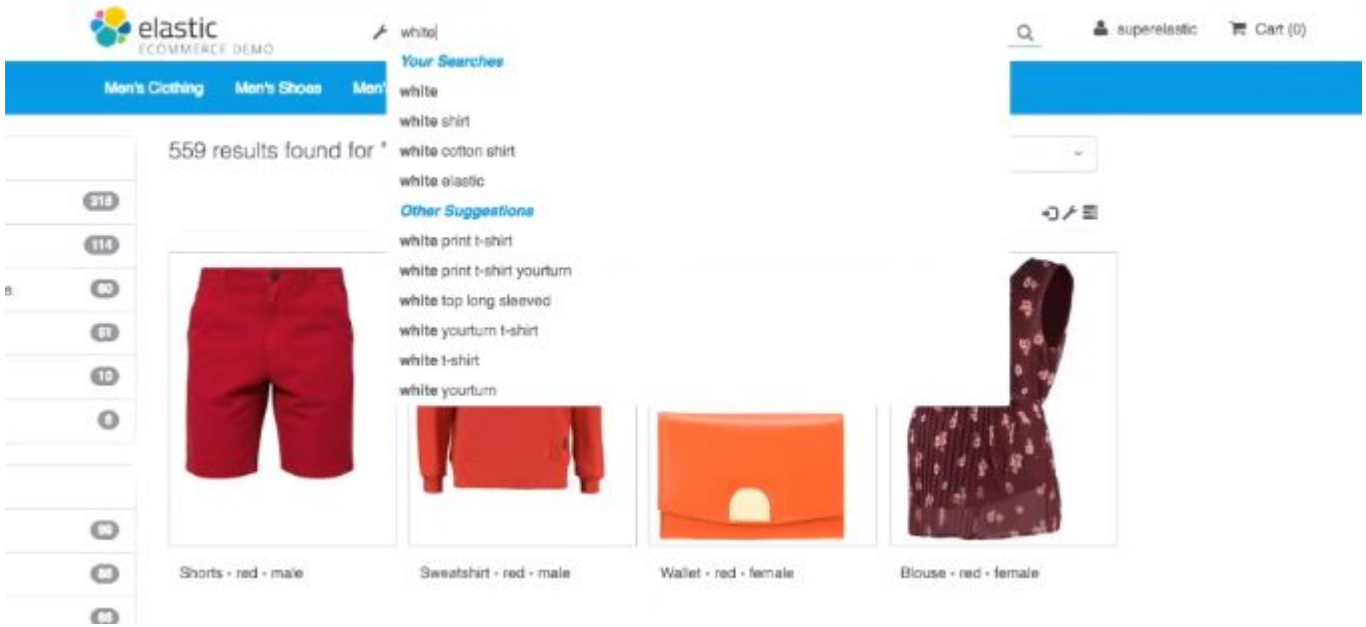
Filtering

```
s = Search()
s = s.filter('terms',
tags=['search', 'python'])
```




```
s = Search(using=es)
    .filter('term',response=404)
    .filter('range',
       timestamp={'gte':'now-5m',
    'lt': 'now'}
    )
```

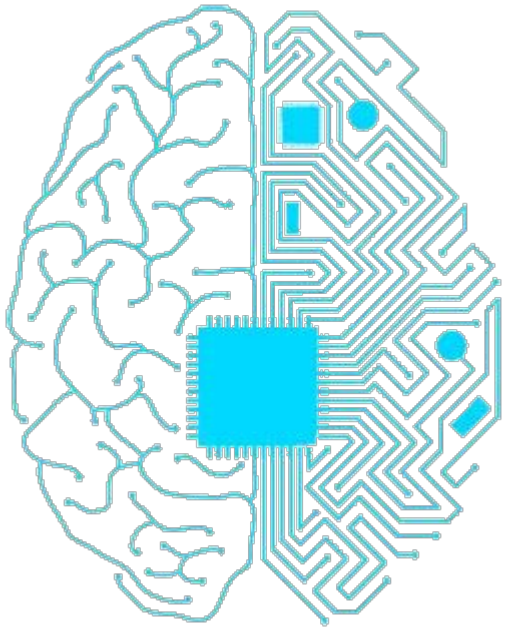
Autocomplete



Autocomplete

```
s = s.suggest('my_suggestion', 'pyhton',
term={'field': 'title'})
```

Some data analysis: NLP



Language Detection

https://github.com/jprante/elasticsearch-langdetect

```
curl -XPOST 'localhost:9200/_langdetect?pretty' -d
'This is a test'

{
  "profile" : "/langdetect/",
  "languages" : [ {
    "language" : "en",
    "probability" : 0.9999971603535163
  } ]
}
```

Preprocessing (Normalization)

```
curl -XGET
"http://localhost:9200/_analyze?analyzer=english" -d'
{
  "text" : "This is a Test."
}'
```

Preprocessing (Normalization)

```
{
  "tokens": [
        {
        "token": "test",
        "start_offset": 10,
        "end_offset": 14,
        "type": "<ALPHANUM>",
        "position": 3
        }
  ]
 }
```

Text Classification

```
es.search(index=INDEX_NAME,
 body = {'query': {
'more_like_this' : {
    'fields' : ['content', 'category'],
    'like' : 'I like python',
    'min_term_freq' : 1,
    'max_query_terms' : 20
    }
   }
})
```

Text Classification

```
from operator import itemgetter
  def get_best_category(response):
      categories = {}
      for hit in response['hits']['hits']:
          score = hit['_score']
          for category in hit['_source']['category']:
              if category not in categories:
                  categories[category] = score
              else:
                  categories[category] += score
      if len(categories) > 0:
          sortedCategories = sorted(categories.items(),
key=itemgetter(1), reverse=True)
          category = sortedCategories[0][0]
      return category
```

Thank You!

Coming soon:
seacom.it/elastic-stack-day-2018