

搜索引擎介绍

@dylanninin

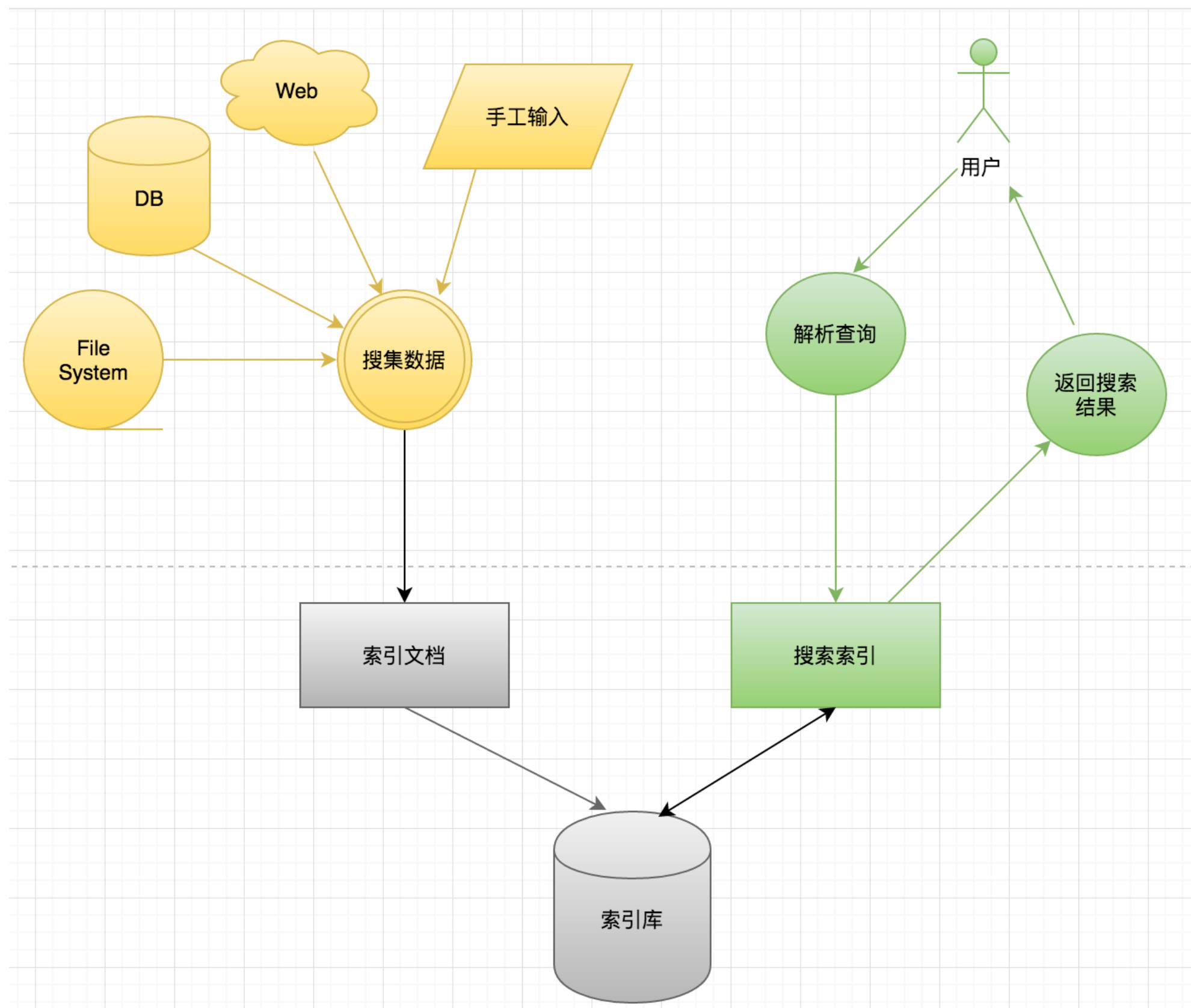
@2016-04-13

触手可及的搜索

- Ctrl + F
- String Search
- Regular Expression
- SQL
- NoSQL
- Search Engine
- Site Search

基本原理

基本原理



ElasticSearch

特点

- 安装配置简单
- 开源
- RESTful API
- 基于 Lucene
- 高可用
- 模式自由
- 分布式
- 面向文档型的设计
- 实时索引



用例

facebook.

 OpenPlay

mozilla


CISCO™

 U B E R

The New York Times

NETFLIX


PERFORM
PROGRESSIVE SPORTS MEDIA

ebay™

GitHub

 stackoverflow

 zendesk®

与数据库类比

ElasticSearch	MongoDB	MySQL
Index 索引	Database 数据库	Database 数据库
Type 类型	Collection 集合	Table 表
Doc 文档	Document 文档	Row 行
Field 字段	Field 字段	Column 列
Mapping 映射	Schema 模式	Schema 模式
Everything Indexed	Index 索引	Index 索引
DSL Query	Query 查询	SQL

演示

http://openplay-staging:19200/_plugin/hq/

搜索实战

搜索案例

所有状态

按地区筛选

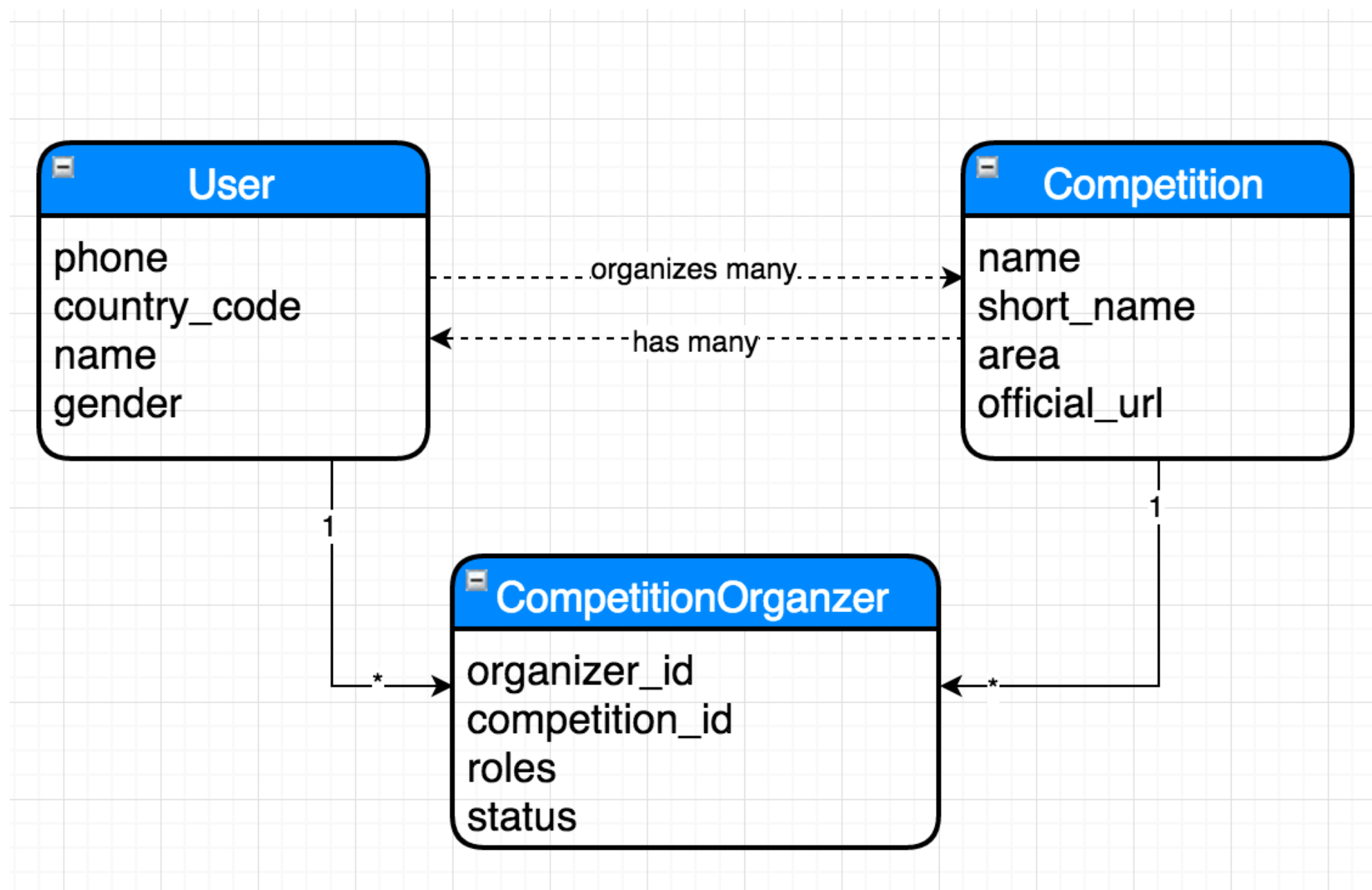
按日期筛选

互联网

OPID	赛事全称	赛事简称	地区	状态	赛事时段
C10014	 互联网足球锦标赛	互锦赛	广东 广州	已结束	2015.11 ~ 2015.12
C10052	 华南互联网兄弟足球队友谊赛	无	广东 广州	已结束	2016.3
C10006	 粤甲联赛测试赛	粤甲联赛测试赛	广东	进行中	无
C10048	 HiALL中欧2016足球超级...	中欧联赛	上海	进行中	无

搜索关键字：OPID, 赛事全称/简称, 赛事组织者
支持排序、分页

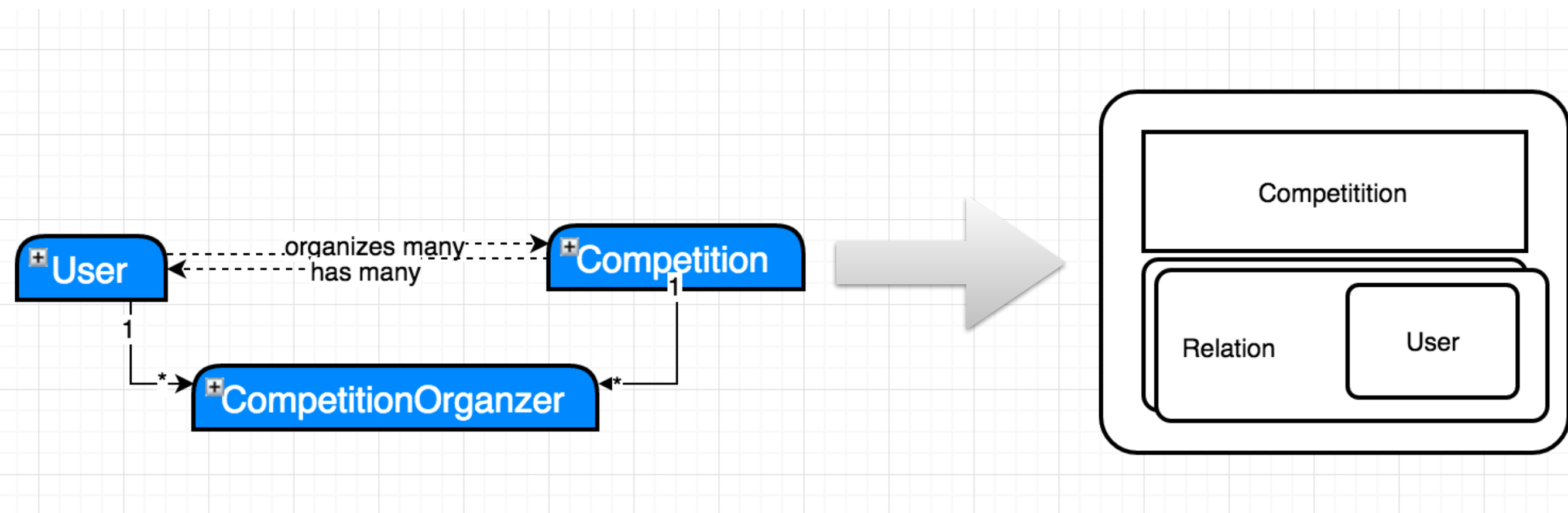
搜索案例



赛事，赛事组织者 实体关系图

数据模型

数据模型

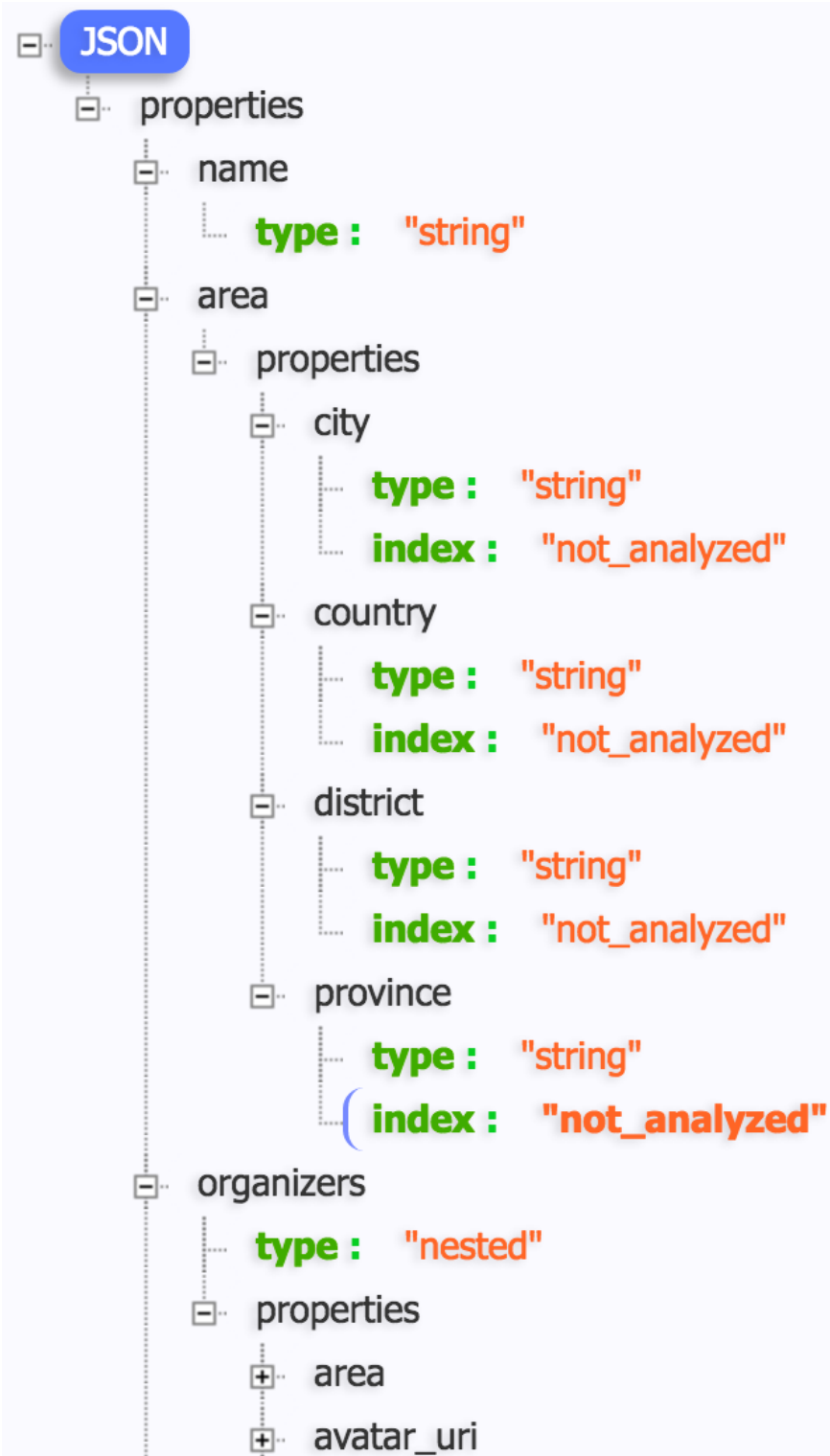


用户、赛事、关系

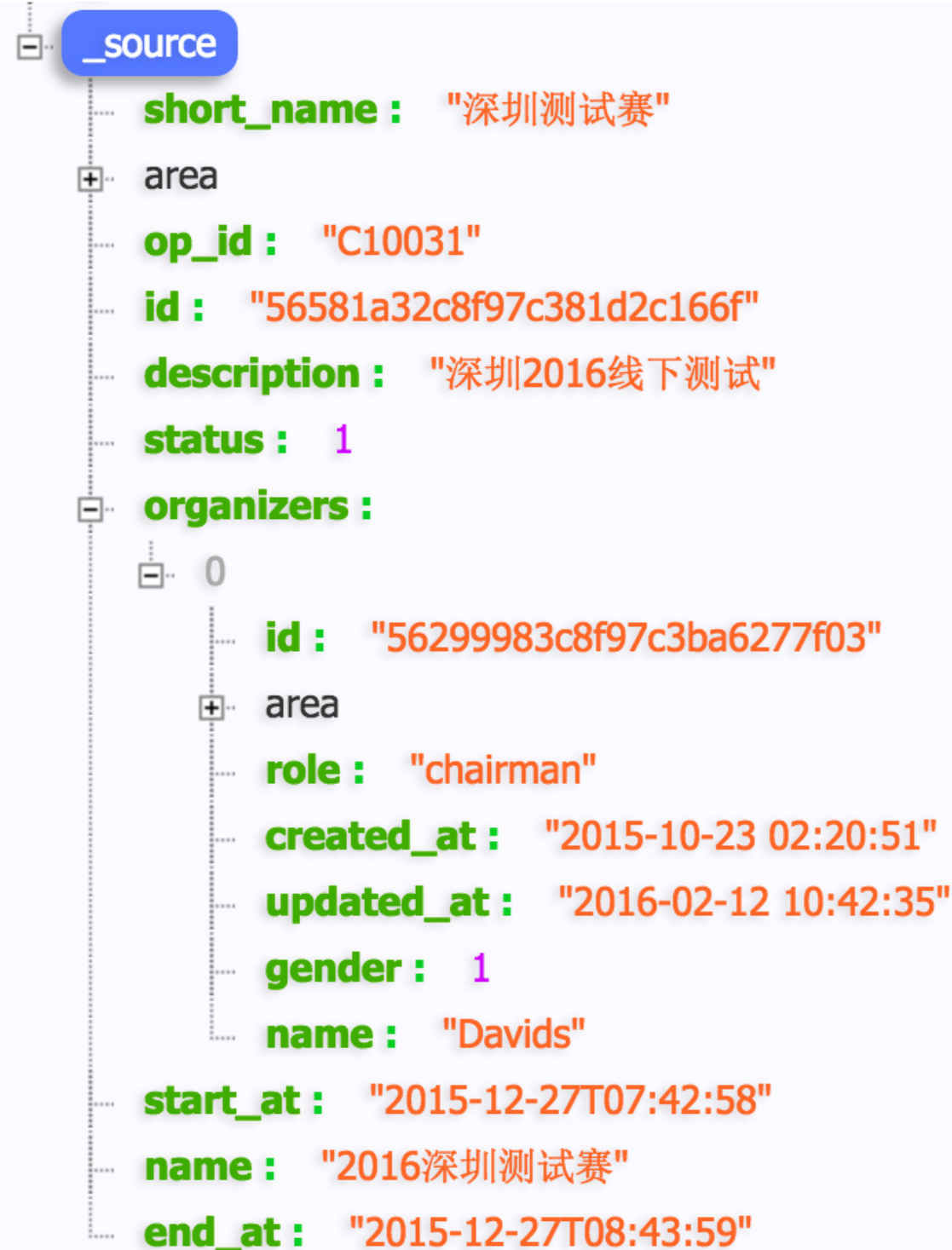
合并到一个 Doc 中

索引数据模型设计

数据模型



mapping



instance

建立索引

建立索引

1. JSON Document

2. _id 为 Document ObjectId

3. RESTful API

POST

{{host}}/op_competition/competition/56b1adc5c8f97c68bd795a2d

Authorization

Headers (1)

Body

Pre-request Script

Tests

form-data

x-www-form-urlencoded

raw

binary

JSON (application/json)

1

{

2

"id": "56b1adc5c8f97c68bd795a2d",

3

"end_at": "2016-02-04T12:00:00",

4

"organizers": [

5

{

6

"birth": "1990-03-14 00:00:00",

7

"gender": 1,

8

"roles": [

9

"statistician"

10

],

11

"created_at": "2015-10-23 02:20:51",

12

"name": "Davids",

13

"updated_at": "2016-04-13 07:03:27",

14

"nationality": "中国",

15

"role": "chairman",

16

"area": {

17

"province": "广东",

18

"district": "海珠",

19

"country": "中国".

Body

Cookies

Headers (2)

Tests

Pretty

Raw

Preview

JSON

1

{

2

"_index": "op_competition",

3

"_type": "competition",

4

"_id": "56b1adc5c8f97c68bd795a2d",

5

"_version": 4,

6

"_shards": {

7

"total": 2,

8

"successful": 1,

9

"failed": 0

10

},

11

"created": true

12

}

智能搜索

智能搜索

界面效果

系列赛事

自由比赛

测试比赛

场馆管理

已结束



中国



2016-02



友谊赛



URL 查询条件

```
/admin/competitions/?q=友谊赛 AND period:Played AND area.country:中国 AND datetime:2016-02&v=2.0
```

OpenPlay query syntax

智能搜索

```
1 {  
2   "query": {  
3     "bool": {  
4       "must": [  
5         {  
6           "bool": {  
7             "should": [  
8               {  
9                 "match": {  
10                  "name": "友谊赛"  
11                }  
12              },  
13              {  
14                "match": {  
15                  "name": "友谊赛"  
16                }  
17              },  
18            ]  
19          }  
20        },  
21        {  
22          "range": {  
23            "start_at": {  
24              "lte": "2016-02-29T16:00:00"  
25            }  
26          }  
27        }  
28      ]  
29    }  
30  },  
31  {  
32    "bool": {  
33      "must": [  
34        {  
35          "range": {  
36            "start_at": {  
37              "lte": "2016-02-29T16:00:00"  
38            }  
39          }  
40        },  
41        {  
42          "range": {  
43            "end_at": {  
44              "gte": "2016-01-31T16:00:00"  
45            }  
46          }  
47        }  
48      ]  
49    }  
50  },  
51  {  
52    "bool": {  
53      "should": [  
54        {  
55          "range": {  
56            "end_at": {  
57              "gte": "2016-01-31T16:00:00"  
58            }  
59          }  
60        },  
61        {  
62          "range": {  
63            "start_at": {  
64              "lte": "2016-02-29T16:00:00"  
65            }  
66          }  
67        }  
68      ]  
69    }  
70  }  
71 }  
72 }
```

ElasticSearch Query DSL

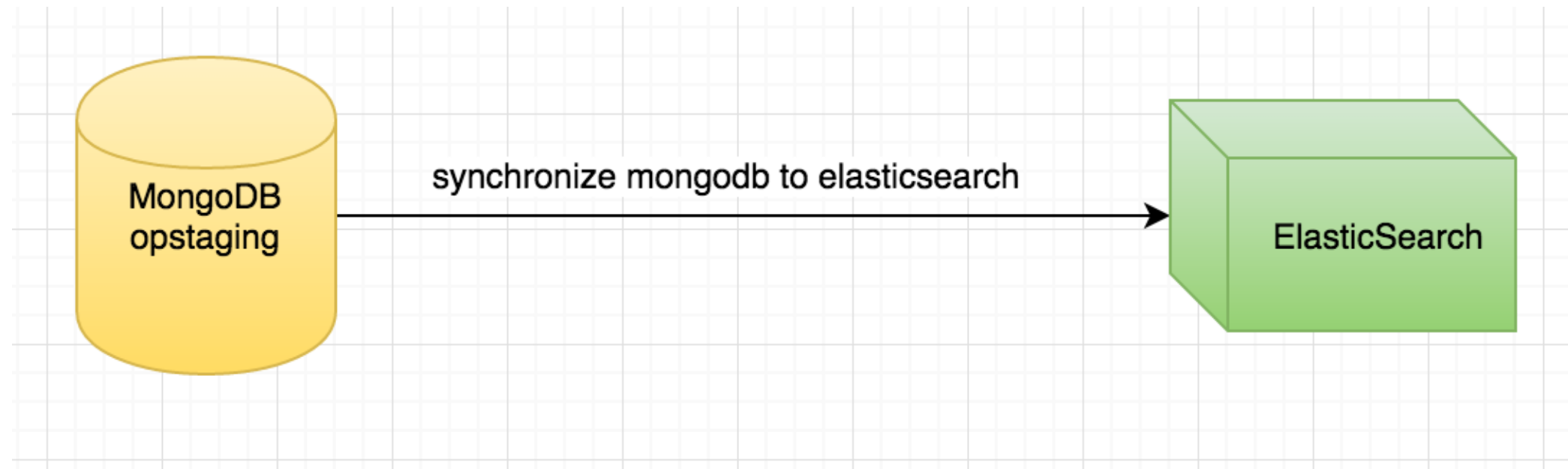
其他功能

1. 排序

2. 分页

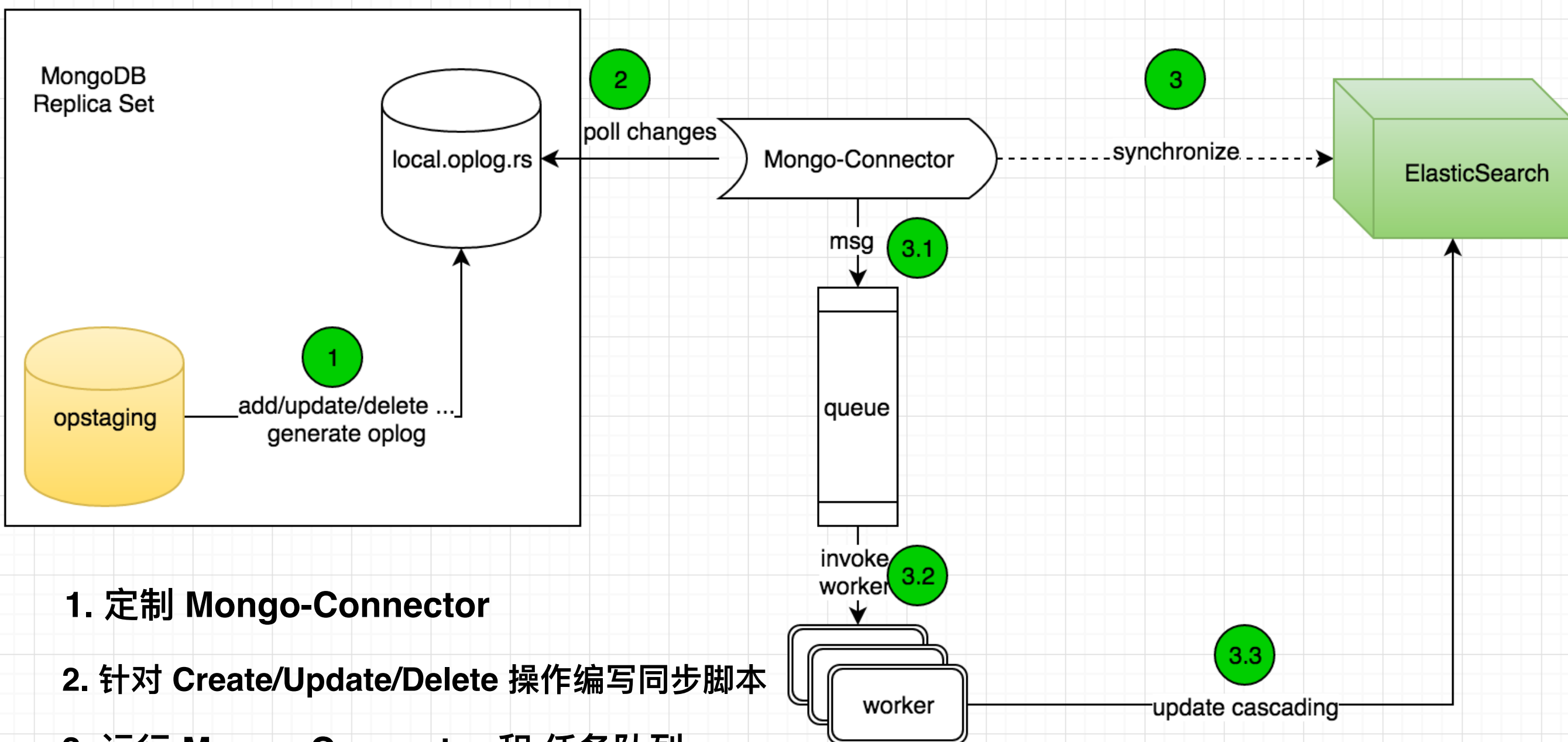
数据同步

数据同步



数据源同步到ElasticSearch

数据同步



1. 定制 Mongo-Connector

2. 针对 Create/Update/Delete 操作编写同步脚本

3. 运行 Mongo-Connector 和 任务队列

OpenPlay example

数据同步

1. 数据更新

- 新增赛事
- 更新赛事
- 删除赛事
- 更新赛事组织者

 competition.py

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4
5  __author__ = 'Dylan'
6
7
8  from pylibs.services.search.models import Competition
9
10
11 def add(data):
12     for _id in collect_object_id(data):
13         Competition.add_by_id(_id)
14
15
16 def update(data):
17     for _id in collect_object_id(data):
18         Competition.update_by_id(_id)
19
20
21 def delete(data):
22     for _id in collect_object_id(data):
23         Competition.delete_by_id(_id)
24
25
26 def update_organizers(data):
27     """
28     赛事组织者发生更新: add/update/delete
29     :param data:
30     :return:
31     """
32     for _id in collect_object_id(data):
33         Competition.update_by_id(_id)
```

OpenPlay example

数据同步

1. 数据更新

2. 级联操作

- 更新用户
- 删除用户

 user.py

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4
5  __author__ = 'Dylan'
6
7  from pylibs.services.search.models import User
8  from pylibs.services.search.models import Competition
9  from pylibs.services.search.models import Match
10 from pylibs.services.search.models import Approval
11
12
13 def add(data):
14     for _id in collect_object_id(data):
15         User.add_by_id(_id)
16
17
18 def update(data):
19     for _id in collect_object_id(data):
20         User.update_by_id(_id)
21         Competition.update_by(filters={'organizers.id': _id})
22         Match.update_by(filters={'statisticians.id': _id})
23         Approval.update_by(filters={'submission.user_id': _id})
24
25
26 def delete(data):
27     for _id in collect_object_id(data):
28         User.delete_by_id(_id)
29         Competition.update_by(filters={'organizers.id': _id})
30         Match.update_by(filters={'statisticians.id': _id})
```

OpenPlay example

高级主题

中文分词

中文分词

jieba_seg.py

```
1  import jieba
2  seg_list = jieba.cut("我来到北京清华大学", cut_all=True)
3  print("Full Mode: " + "/ ".join(seg_list))
4  # Full Mode: 我/ 来到/ 北京/ 清华/ 清华大学/ 华大/ 大学
5
6  seg_list = jieba.cut("我来到北京清华大学", cut_all=False)
7  print("Default Mode: " + "/ ".join(seg_list))
8  # Default Mode: 我/ 来到/ 北京/ 清华大学
9
10 seg_list = jieba.cut_for_search("我来到北京清华大学")
11 print("Search Mode: " + "/ ".join(seg_list))
12 # Search Mode: 我/ 来到/ 北京/ 清华/ 华大/ 大学/ 清华大学
```

jieba 分词

中文分词

1. 内置分词器

GET ⌵

{{host}}/op_user/_analyze?analyzer=standard&text=我来到北京清华大学

Authorization

Headers (1)

Body

Pre-request Script

Tests

Type

No Auth ⌵

Body

Cookies

Headers (7)


Tests

Pretty

Raw

Preview

JSON ⌵



1

{

2

"tokens": [

3

{

4

"token": "我",

5

"start_offset": 0,

6

"end_offset": 1,

7

"type": "<IDEOGRAPHIC>",

8

"position": 0

9

},

10

{

11

"token": "来",

12

"start_offset": 1,

13

"end_offset": 2,

14

"type": "<IDEOGRAPHIC>",

15

"position": 1

16

},

17

{

18

"token": "到",

19

"start_offset": 2,

20

"end_offset": 3,

21

"type": "<IDEOGRAPHIC>",

22


"position": 2

23

},

中文分词

1. ik分词器

GET 

{{host}}/op_user/_analyze?analyzer=ik&text=我来到北京清华大学

Authorization


Headers (1)

Body

Pre-request Script

Tests

Type

No Auth 

Body

Cookies


Headers (7)


Tests

Pretty

Raw

Preview

JSON 



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

{

"tokens": [

{

"token": "我",

"start_offset": 0,

"end_offset": 1,

"type": "CN_CHAR",

"position": 0

},

{

"token": "来到",

"start_offset": 1,

"end_offset": 3,

"type": "CN_WORD",

"position": 1

},

{

"token": "北京",

"start_offset": 3,

"end_offset": 5,

"type": "CN_WORD",

"position": 2

},

}

中文分词

1. 基于规则

字符串匹配，即扫描字符串，如果发现字符串的子串和词相同，就算匹配

“正向/反向最大匹配”，“长词优先”等策略

$O(n)$ 时间复杂度，实现简单，效果尚可

对歧义和未登录词处理不好

2. 基于概率

基于人工标注的词性和统计特征，对中文进行建模，早出概率最大的分词结果

能很好处理歧义和未登录词问题，效果比前一类效果好

需要大量的人工标注数据，以及较慢的分词速度

倒排索引

建立索引

1. The quick brown fox jumped over the lazy dog
2. Quick brown foxes leap over lazy dogs in summer

Search: brown quick

建立索引

1. The quick brown fox jumped over the lazy dog

2. Quick brown foxes leap over lazy dogs in summer

Term	Doc_1	Doc_2

Quick		X
The	X	
brown	X	X
dog	X	
dogs		X
fox	X	
foxes		X
in		X
jumped	X	
lazy	X	X
leap		X
over	X	X
quick	X	
summer		X
the	X	

inverted index

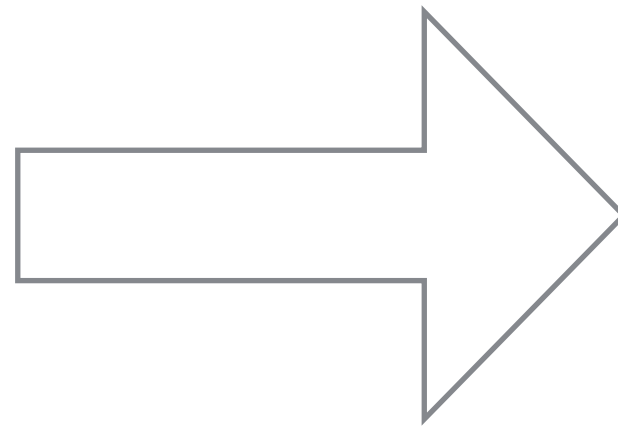
建立索引

1. The quick brown fox jumped over the lazy dog

2. Quick brown foxes leap over lazy dogs in summer

Term	Doc_1	Doc_2

Quick		X
The	X	
brown	X	X
dog	X	
dogs		X
fox	X	
foxes		X
in		X
jumped	X	
lazy	X	X
leap		X
over	X	X
quick	X	
summer		X
the	X	



Term	Doc_1	Doc_2

brown	X	X
quick	X	

Total	2	1

brown quick

inverted index

建立索引

1. The quick brown fox jumped over the lazy dog

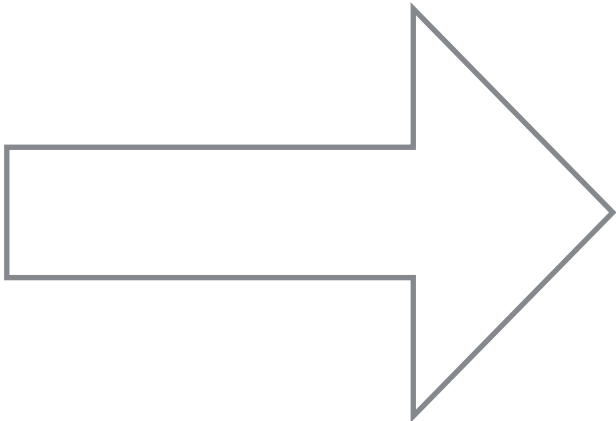
2. Quick brown foxes leap over lazy dogs in summer

Term	Doc_1	Doc_2

Quick		X
The	X	
brown	X	X
dog	X	
dogs		X
fox	X	
foxes		X
in		X
jumped	X	
lazy	X	X
leap		X
over	X	X
quick	X	
summer		X
the	X	

inverted index		

Quick, quick
dog, dogs
jumped, leap



Term	Doc_1	Doc_2

brown	X	X
dog	X	X
fox	X	X
in		X
jump	X	X
lazy	X	X
over	X	X
quick	X	X
summer		X
the	X	X

inverted index		

数据聚合

数据聚合

数据总览	
赛事	63 ^
赛事地区	<div>中国 (49) 中国台湾 (2) 中国香港 (2)</div> <div>广东 (35) 上海 (7) 四川 (4) 台湾 (2) 香港 (2) 北京 (1) 重庆 (1)</div> <div>广州 (28) 上海 (7) 深圳 (4) 成都 (2) 广安 (1) 北京 (1) 重庆 (1)</div> <div>--国外--</div> <div>阿尔及利亚 (1)</div>

数据聚合

1. aggs 语法

2. 自定义结构

POST

{{host}}/op_user/user/_search

AuthorizationHeaders (1)BodyPre-request ScriptTests

form-data

x-www-form-urlencoded

raw

binary

JSON (application/json)

1

{

2

"aggs" : {

3

"country" : {

4

"terms" : { "field" : "area.country" },

5

"aggs" : {

6

"province" : {

7

"terms" : { "field" : "area.province" },

8

"aggs" : {

9

"city" : {

10

"terms" : { "field" : "area.city" }

11

}

12

}

13

}

14

}

15

}

16

}

17

}

BodyCookiesHeaders (2)Tests

PrettyRawPreview

JSON

829

{

830

"key": "中国台湾",

831

"doc_count": 2,

832

"province": {

833

"doc_count_error_upper_bound": 0,

834

"sum_other_doc_count": 0,

835

"buckets": [

836

{

837

"key": "台湾",

838

"doc_count": 2,

839

"city": {

840

"doc_count_error_upper_bound": 0,

841

"sum_other_doc_count": 0,

842

"buckets": [

搜索结果高亮

搜索结果高亮

openplay 数据统计



All

News

Images

Videos

Maps

More ▼

Search tools

About 5,580 results (0.64 seconds)

TCSH 2015 | OpenPlay给业余足球员专业统计数据- 动点科技

cn.technode.com/post/2015-06.../tcs2015-openplay/ ▼ Translate this page

Jun 9, 2015 - OpenPlay 是国内体育社区洋葱圈开始的新业务，旨在为业余足球员提供相对专业的技术统计，让石家庄梅西、徐汇伊布这样的称号有可靠的数据 ...

OpenPlay - 为任何足球比赛提供精确、详尽的技术统计数据 ...

next.36kr.com/posts/29327 ▼ Translate this page

5 days ago - OpenPlay - 为任何足球比赛提供精确、详尽的技术统计数据- NEXT.

OpenPlay: 首页

op.ai/ ▼ Translate this page

皮球还在空中飞行，数据已在空中传输；应声入网，立即呈现. 精确 ... 告诉我们你的赛事有多少参赛球队、比赛总场次和开赛时间，抢先体验OpenPlay 技术统计！



OpenPlay app下载|OpenPlay数据统计软件下载v1.0.4 ...

www.gezila.com › ... › Android软件 › 生活应用 ▼ Translate this page

2 days ago - OpenPlay app 记录你在球场上的每一项数据，让你拥有如职业球员般的数据统计。无论是向朋友炫耀你的进球数、传球成功率和球员评分，还是根据 ...

搜索结果高亮

OpenPlay 数据统计



全部 新聞 圖片 影片 地圖 更多 ▾ 搜尋工具

約 5,780 項搜尋結果 (0.56 秒)

TCSH 2015 给业余足球员专业统计数据- 动点科技

cn.technode.com.cn/2015-06-09/tcsh2015-openplay/ ▾ 轉為繁體網頁

2015年6月9日 - OpenPlay 是国内体育社区洋葱圈开始的新业务，旨在为业余足球员提供相对专业的技术统计，让石家庄梅西、徐汇伊布这样的称号有可靠的数据 ...

OpenPlay app下载|OpenPlay数据统计软件下载v1.0.4 ...

ts Console Sources Network Timeline Profiles Resources Security Audits Adblock Plus EditThisCookie

```
▼<div>
  ▶<div class="f kv _SWb" style="white-space:nowrap">...</div>
  ▼<span class="st">
    <span class="f">2015年6月9日 - </span>
    <em>OpenPlay</em> == $0
    " 是国内体育社区洋葱圈开始的新业务，旨在为业余足球员提供相对专业的技术"
    <em>统计</em>
    "，让石家庄梅西、徐汇伊布这样的称号有可靠的"
    <em>数据</em>
    "&nbsp;&nbsp;&nbsp;..."
  </span>
</div>
</div>
</div>
<!--n-->
</div>
```

搜索结果高亮

1. highlight 语法

2. 自定义标签

POST ▼ {{host}}/op_user/user/_search

Authorization Headers (1) **Body** ● Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary **JSON (application/json)** ▼

```
1 {
2   "query": {
3     "match": {
4       "name": "张祥贵"
5     }
6   },
7   "highlight": {
8     "fields": {
9       "name": {
10      }
11    }
12  }
13 }
```

Body Cookies Headers (2) Tests

Pretty Raw Preview **JSON** ▼

```
57   "nationality": "中国",
58   "name": "张祥贵"
59 },
60 "highlight": {
61   "name": [
62     "<em>张</em><em>祥</em><em>贵</em>"
63   ]
64 }
```

搜索建议

搜索建议

openpl



openplex

open plan

openplacement

openplay

openplotter

openpli

openplc

openplate

openplm

openplane

Google Search

I'm Feeling Lucky

搜索建议

1. Mapping 定义suggest 数据源

PUT ▾

{{host}}/music

Authorization

Headers (1)

Body ●

Pre-request Script

Tests

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

JSON (application/json) ▾

1 ▾

2 ▾

3 ▾

4 ▾

5

6

7 ▾

8

9

10

11

12

13

14

15

```
{
  "song" : {
    "properties" : {
      "name" : {
        "type" : "string"
      },
      "suggest" : {
        "type" : "completion",
        "analyzer" : "simple",
        "search_analyzer" : "simple",
        "payloads" : true
      }
    }
  }
}
```


搜索建议

1. Mapping 定义suggest 数据源

2. 添加数据

POST ▾

{{host}}/music/song/1

Authorization

Headers (1)

Body ●

Pre-request Script

Tests

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

JSON (application/json) ▾

1 ▾

{

2

3 ▾

4 "name" : "Nevermind",

5 "suggest" : {

6 "input": ["Nevermind", "Nirvana"],

7 "output": "Nirvana - Nevermind",

8 "payload" : { "artistId" : 1 },

9 "weight" : 10

8 }

9 }

搜索建议

1. Mapping 定义sugge

2. 添加数据

3. 搜索建议

POST {{host}}/music/_suggest

AuthorizationHeaders (1)BodyPre-request ScriptTests

form-data

x-www-form-urlencoded

raw

binary

JSON (application/json)

1

2

3

4

5

6

7

8

{

"song-suggest" : {

"text" : "n",

"completion" : {

"field" : "suggest"

}

}

}

BodyCookiesHeaders (2)Tests

Pretty

Raw

Preview

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

{

"_shards": {

"total": 5,

"successful": 5,

"failed": 0

},

"song-suggest": [

{

"text": "n",

"offset": 0,

"length": 1,

"options": [

{

"text": "Nirvana - Nevermind",

"score": 10,

"payload": {

"artistId": 1

}

}

]

}

]

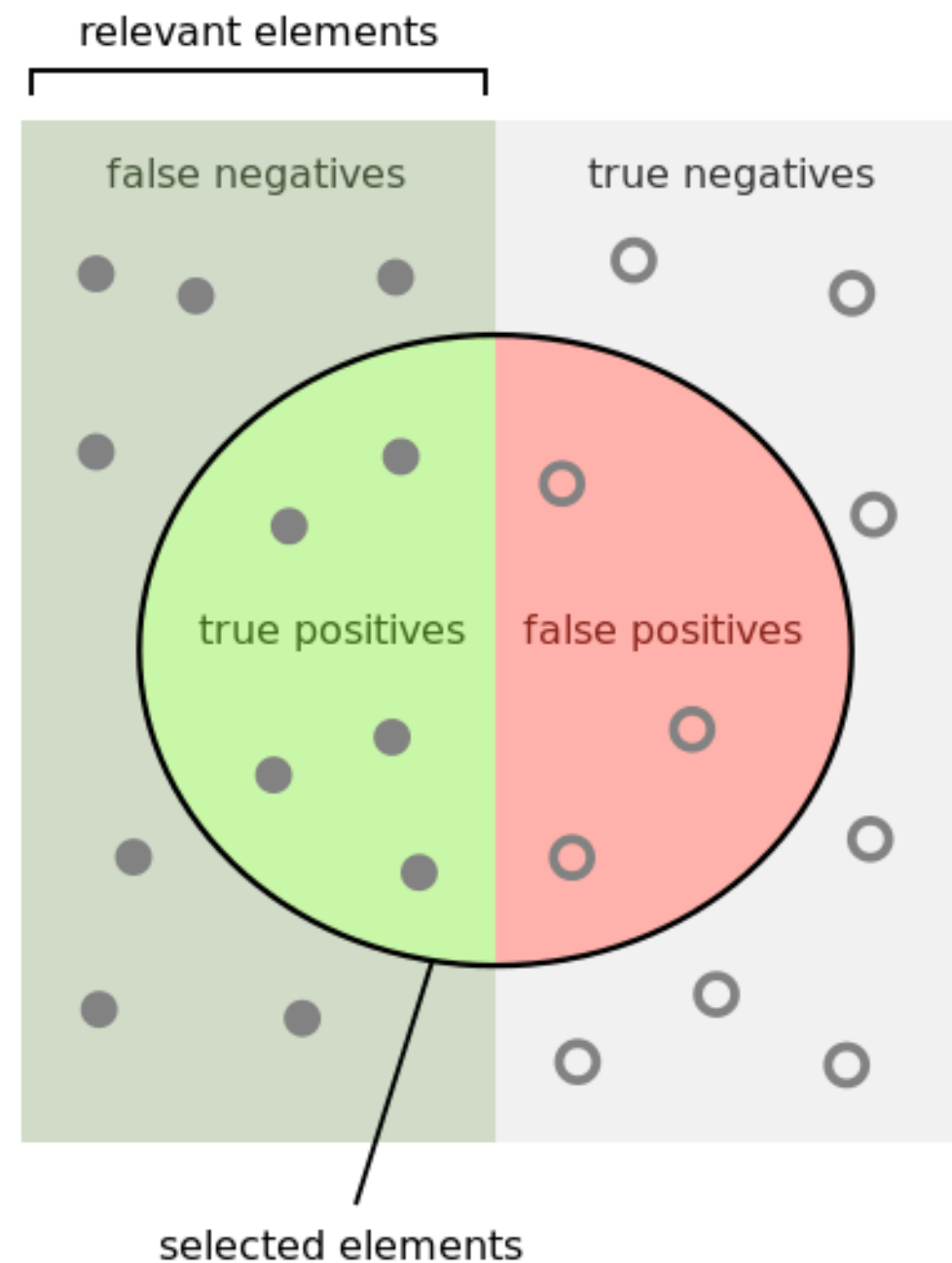
}

搜索引擎难点

准确度

1. 精确度 (precision)

2. 召回率 (recall)



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

响应速度

openplay 数据统计



All

News

Images

Videos

Maps

More ▼

Search tools

About 5,550 results (0.56 seconds)

[TCSH 2015 | OpenPlay给业余足球队专业统计数据- 动点科技](#)

cn.technode.com/post/2015-06.../tcs2015-openplay/ ▼ [Translate this page](#)

Jun 9, 2015 - **OpenPlay** 是国内体育社区洋葱圈开始的新业务，旨在为业余足球队提供相对专业的技术**统计**，让石家庄梅西、徐汇伊布这样的称号有可靠的**数据** ...

响应速度

数据总览		
赛事	63	▼
比赛	1,154	▼
球队	307	▲
球队地区	<div>中国 (263) 中国香港 (14)</div> <div>广东 (159) 四川 (55) 上海 (20) 香港 (14) 广东省 (8) 湖南 (7) 上海市 (5) 重庆 (4) 北京 (3) 江西 (1)</div> <div>广州 (105) 成都 (41) 深圳 (34) 上海 (20) 广州市 (7) 九龙 (6) 株洲 (5) 上海市 (5) 重庆 (4) 北京 (3) 绵阳 (2) 自贡 (2) 清远 (2) 南充 (2) 深圳市 (1) 广安 (1) 湘潭 (1) 邵阳 (1) 遂宁 (1) 梅州 (1) 珠海 (1) 萍乡 (1) 韶关 (1) 香港岛 (1)</div> <div>--国外--</div> <div>安道尔 (3) 刚果 (1) 日本 (1) 智利 (1) 澳大利亚 (1) 爱尔兰 (1)</div>	
球队成员	4,077	▼
组织者	30	▼

```
160414 11:39:59 base:63] GET http://localhost:9200/op_competition/competition/_search [status:200 request:0.016s]
160414 11:39:59 base:63] GET http://localhost:9200/op_team/team/_search [status:200 request:0.010s]
160414 11:39:59 base:63] GET http://localhost:9200/op_user/user/_search [status:200 request:0.013s]
160414 11:39:59 base:63] GET http://localhost:9200/op_user/user/_search [status:200 request:0.006s]
160414 11:39:59 statistics:55] 执行时间：0.034908000000000149
160414 11:39:59 web:1908] 200 GET /admin/overview/statistics/?v=2.0 (127.0.0.1) 100.89ms
```

实时性

1. 实时建立索引

2. 实时查询最新的数据

QA

参考

- Wiki: Search Engine
- <https://www.elastic.co/guide/index.html>
- ElasticSearch 权威指南
- ElasticSearch 中文发行版
- Lucene
- Mongo-Connector
- Jieba 分词
- 盘古分词
- Inverted Index
- Wiki: Precision Recall
- Information Retrieval and Web Search