

link: null
title: 珠峰架构师成长计划
description: http://localhost:9200/
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=48 sentences=33, words=370

1. 全文搜索

- 开源的 Elasticsearch（以下简称 Elastic）是目前全文搜索引擎的首选。
- 它可以快速地储存、搜索和分析海量数据

2. 安装

1. 安装jdk(<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>)
2. 安装elasticsearch(<https://www.elastic.co/downloads/elasticsearch>)

3. 访问

<http://localhost:9200/> (<http://localhost:9200/>)

```
{
  "name" : "PC-201704292335",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "pb4rTAeoSxyLgJGtyz4fAg",
  "version" : {
    "number" : "5.6.10",
    "build_hash" : "b727a60",
    "build_date" : "2018-06-06T15:48:34.860Z",
    "build_snapshot" : false,
    "lucene_version" : "6.6.1"
  },
  "tagline" : "You Know, for Search"
}
```

- C:\ProgramData\Elastic\Elasticsearch\config\elasticsearch.yml 配置文件

4. 基本概念

4.1 节点和集群

- Elastic 本质上是一个分布式数据库，允许多台服务器协同工作，每台服务器可以运行多个 Elastic 实例
- 单个 Elastic 实例称为一个节点node。一组节点构成一个集群cluster

4.2 索引

- Elastic 会索引所有字段，经过处理后写入一个反向索引（Inverted Index）。查找数据的时候，直接查找该索引
- Elastic 数据管理的顶层单位就叫做 Index（索引），Index（即数据库）的名字必须是小写

4.3 文档

- Index 里面单条的记录称为 Document（文档）
- 许多条 Document 构成了一个 Index
- Document 使用 JSON 格式表示

4.4 类型(Type)

- 文档可以分组，比如学生这个Index里面，可以按性别分组(男生一组，女生一组)，也可以按省份分组(北京和广东)
- 这种分组就叫做类型，它是虚拟的逻辑分组，用来过滤文档
- 不同的类型应该有相似的结构
- 每个 Index 包含一个 Type，7.x 版将会彻底移除 Type

5. 操作Index

5.1 创建索引

```
curl -X PUT 'http://localhost:9200/student'
```

- 不能有请求体

5.2 删除索引

```
curl -X DELETE 'http://localhost:9200/student'
```

6. 数据操作

6.1 新增文档

```
curl -X PUT 'http://localhost:9200/student/city/1' -d`
{
  "name": "张三",
  "age": 5,
  "city": "北京"
}
```

```
curl -X POST 'http://localhost:9200/student/city' -d`
{
  "name": "赵六",
  "age": 7,
  "city": "江苏"
}
```

6.2 查看文档

```
curl 'http://localhost:9200/student/city/1'
```

6.3 更新记录

```
curl -X PUT 'http://localhost:9200/student/city/1' -d `
{
  "name": "张三2",
  "age": 55,
  "city": "北京2"
}
```

6.4 删除文档

```
curl -X DELETE 'http://localhost:9200/student/city/1'
```

7. 数据查询

7.1 查询全部

```
curl 'http://localhost:9200/student/city/_search'
```

7.2 全文搜索

```
curl 'http://localhost:9200/student/city/_search' -d `
{
  "query": { "match" : { "name" : "李" }},
  "size": 1,
  "from": 1
}
`
```

7.3 OR

```
curl 'http://localhost:9200/student/city/_search' -d `
{
  "query": { "match" : { "name" : "赵 李" }}
}
`
```

7.4 AND

```
curl 'http://localhost:9200/student/city/_search' -d `
{
  "query": {
    "bool": {
      "must": [
        { "match": { "name": "赵" } },
        { "match": { "name": "六" } }
      ]
    }
  }
}
`
```

8. node中如何用

```
var elasticsearch = require('elasticsearch');
var client = new elasticsearch.Client({
  host: 'localhost:9200',
  log: 'trace'
});

(async function () {
  let name = Date.now();
  let id = Date.now();
  const created = await client.create({
    index: 'student',
    type: 'city',
    id,
    body: {
      name,
      age: 10
    }
  });
  console.log(created);
  const updated = await client.update({
    index: 'student',
    type: 'city',
    id,
    body: {
      doc: {
        name: name,
        age: 101
      }
    }
  });
  console.log(updated);

  console.log(deleted);
})().then(ret => console.log(ret), err => console.log(err));
```

9. 参考

- [elastic \(https://www.elastic.co/cn/\)](https://www.elastic.co/cn/)
- [installation \(https://www.elastic.co/downloads/elasticsearch\)](https://www.elastic.co/downloads/elasticsearch)
- [elasticsearch \(https://github.com/elastic/elasticsearch-js\)](https://github.com/elastic/elasticsearch-js)
- [api \(https://www.elastic.co/guide/en/elasticsearch/client/javascript-api/current/api-reference.html#api-delete\)](https://www.elastic.co/guide/en/elasticsearch/client/javascript-api/current/api-reference.html#api-delete)