



# ElasticSearch

# 目录 Contents

- ◆ 初识 Elasticsearch
- ◆ 安装 Elasticsearch
- ◆ Elasticsearch 核心概念
- ◆ 操作 Elasticsearch
- ◆ Elasticsearch JavaAPI

# 初识

---

# ElasticSearch

- 数据库存在的问题
- 倒排索引
- ElasticSearch概念

# 初识 Elasticsearch

ElasticSearch是一个搜索服务器



说到搜索，大家能想到什么？



# 初识 Elasticsearch

ElasticSearch是一个搜索服务器

搜索就是查询

select \* from xxx



关系型数据库



## 数据库查询的问题

id	title	sell_point
909245	酷派 8076D 咖啡棕 移动3G手机 双	4.0英寸屏幕, 5
912107	创维 (Skyworth) LED 42E5DHR 42英	电视二代: 二代新品!! 智
915676	联想 P780 极速版 深邃黑 联通3G	待机王, 5吋HD
917460	华为 P6 (P6-C00) 黑 电信3G手机	经典旗舰! 雅然
917461	华为 P6 (P6-C00) 白 电信3G手机	经典旗舰, 万人
917770	华为 P6-C00 电信3G手机 (粉色) CDMA2000/GSM 双模双待双通	情人节神器! 粉

1亿条



查询title中包含 '手机' 的信息?

```
SELECT * FROM goods WHERE title LIKE '%手机%';
```



如果使用模糊查询, 左边有通配符, 不会走索引, 会全表扫描, 性能低

## 数据库查询的问题

	id	title	sell_point
<input type="checkbox"/>	909245	酷派 8076D 咖啡棕 移动3G手机 双卡双待	4.0英寸屏幕, 5
<input type="checkbox"/>	912107	创维 (Skyworth) LED 42E5DHR 42英寸 海量影视资源智能网络平板液晶电视二代:	二代新品!! 智
<input type="checkbox"/>	915676	联想 P780 极速版 深邃黑 联通3G手机 双卡双待	待机王, 5吋HD
<input type="checkbox"/>	917460	华为 P6 (P6-C00) 黑 电信3G手机 双卡双待双通	经典旗舰! 雅然
<input type="checkbox"/>	917461	华为 P6 (P6-C00) 白 电信3G手机 双卡双待双通	经典旗舰, 万人
<input type="checkbox"/>	917770	华为 P6-C00 电信3G手机 (粉色) CDMA2000/GSM 双模双待双通	情人节神器! 粉



查询title中包含‘华为手机’的信息?

```
SELECT * FROM goods WHERE title LIKE '%华为手机%';
```



关系型数据库提供的查询, 功能太弱

## 数据库查询的问题



性能低



功能弱

倒排索引



## 倒排索引



请说出包含“前”的诗句？



请背诵《静夜思》

床前明月光，疑是地上霜。  
举头望明月，低头思故乡。

## 正向索引

Key	Value
《静夜思》	床前明月光...
《春晓》	春眠不觉晓...
《水调歌头》	明月几时有？把酒问青天...

## 倒排索引

“床前明月光” --> 分词

将一段文本按照一定的规则，拆分为不同的**词条** (term)

### 反向 (倒排) 索引

Key (term)	Value
床	床前明月光...
前	床前明月光...
床前	床前明月光...
明月	床前明月光...
月	床前明月光...
光	床前明月光...
月光	床前明月光...

“明月几时有”

Key (term)	Value
床	床前明月光...
前	床前明月光...
床前	床前明月光...
明月	床前明月光..., 明月几时有...
月	床前明月光..., 明月几时有...
光	床前明月光...
月光	床前明月光...
几时	明月几时有...
有	明月几时有...

## 倒排索引

Key (term)	Value
床	床前明月光...
前	床前明月光...
床前	床前明月光...
明月	床前明月光...明月几时有...
月	床前明月光...明月几时有...
光	床前明月光...
月光	床前明月光...
几时	明月几时有...
有	明月几时有...



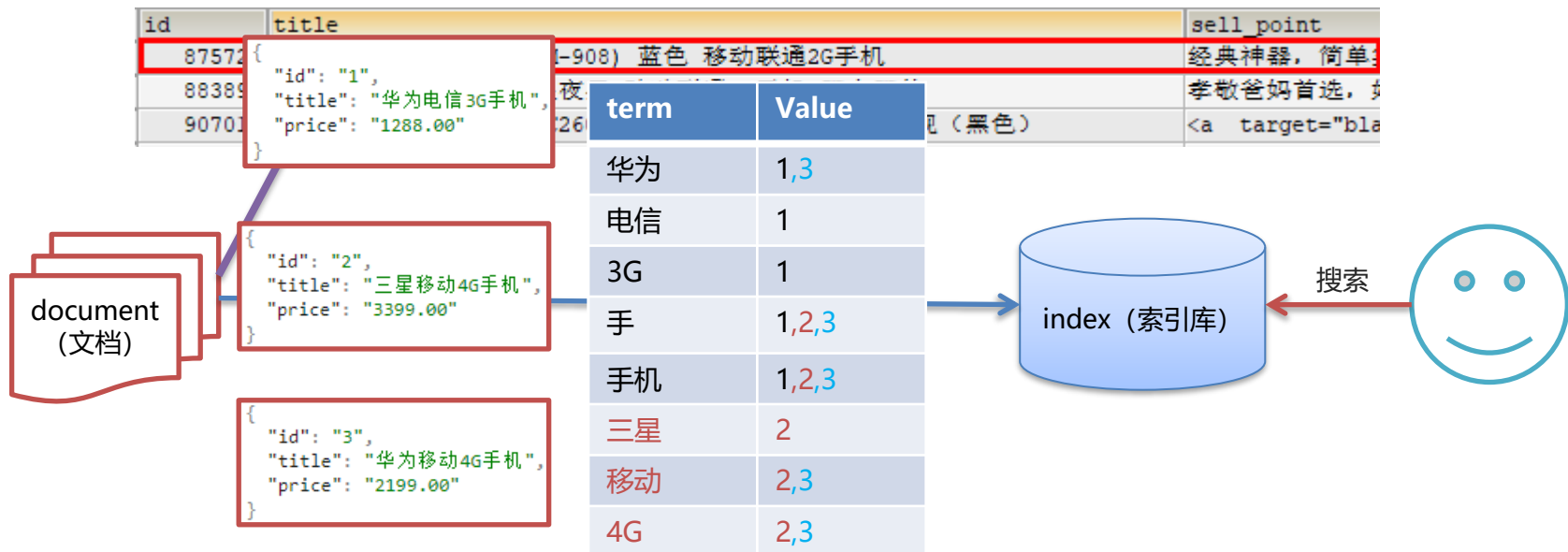
Key (term)	Value
床	《静夜思》
前	《静夜思》
床前	《静夜思》
明月	《静夜思》，《水调歌头》
月	《静夜思》，《水调歌头》
光	《静夜思》
月光	《静夜思》
几时	《水调歌头》
有	《水调歌头》

## 倒排索引

倒排索引：将各个文档中的内容，进行分词，形成词条。然后记录词条和数据的唯一标识（id）的对应关系，形成的产物。

Key (term)	Value
床	《静夜思》
前	《静夜思》
床前	《静夜思》
明月	《静夜思》，《水调歌头》
月	《静夜思》，《水调歌头》
光	《静夜思》
月光	《静夜思》
几时	《水调歌头》
有	《水调歌头》

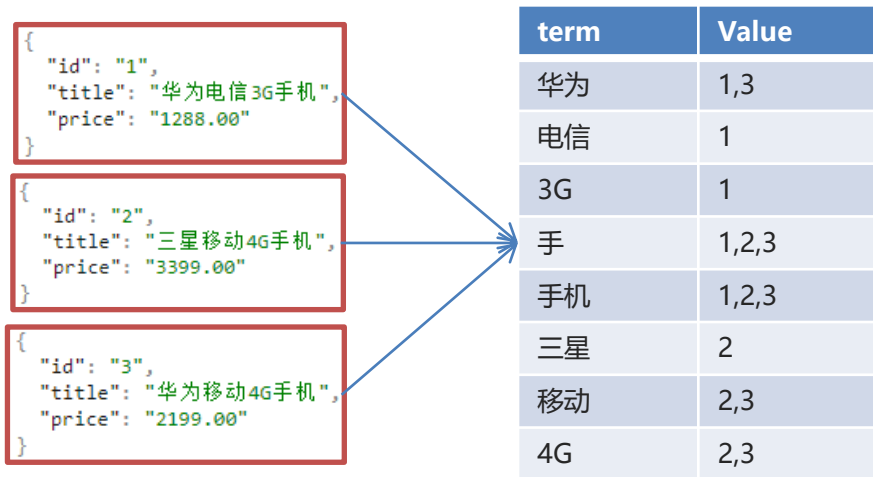
## ElasticSearch数据的存储和搜索原理



## ElasticSearch数据的存储和搜索原理

### 数据库查询存在的问题：

1. 性能低：使用模糊查询，左边有通配符，不会走索引，会全表扫描，性能低
2. 功能弱：如果以“华为手机”作为条件，查询不出来数据



### 1. 使用“手机”作为关键字查询

生成的倒排索引中，词条会排序，形成一颗树形结构，提升词条的查询速度

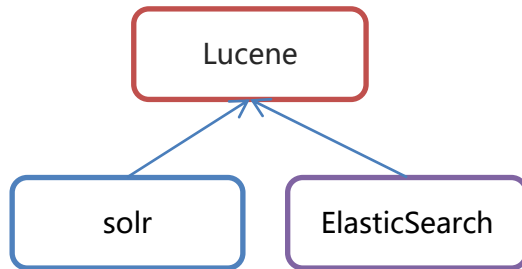
### 2. 使用“华为手机”作为关键字查询

华为 : 1,3

手机 : 1,2,3

## ElasticSearch概念

- Elasticsearch是一个基于Lucene的搜索服务器
- 是一个分布式、高扩展、高实时的搜索与数据分析引擎
- 基于RESTful web接口
- Elasticsearch是用Java语言开发的，并作为Apache许可条款下的开放源码发布，是一种流行的企业级搜索引擎
- 官网：<https://www.elastic.co/>



### 应用场景

- 搜索：海量数据的查询
- 日志数据分析
- 实时数据分析

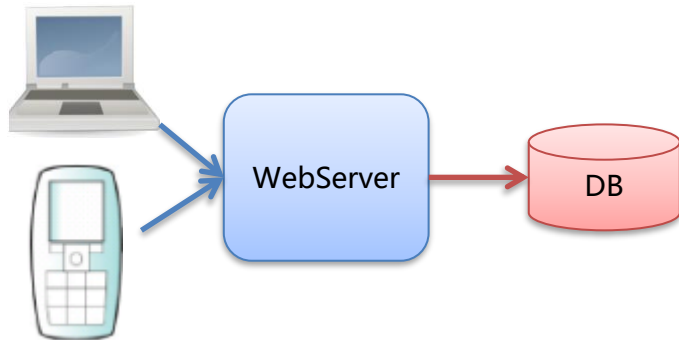


## ElasticSearch概念

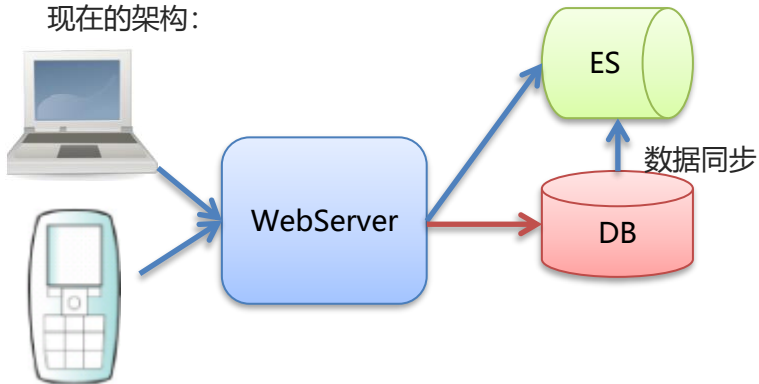
- MySQL有事务性,而ElasticSearch没有事务性,所以你删了的数据是无法恢复的。
- ElasticSearch没有物理外键这个特性,如果你的数据强一致性要求比较高,还是建议慎用

**ElasticSearch和MySQL分工不同, MySQL负责存储数据, ElasticSearch负责搜索数据。**

以前的架构:



现在的架构:





## 小结

- 数据库查询存在的问题
  - 性能低
  - 功能弱
- 倒排索引
  - 将文档进行分词，形成词条和id的对应关系。
- Elasticsearch概念及作用
  - 搜索服务器
  - 高性能，功能强大
  - 不是替换mysql，是与mysql分工不同，es做搜索，mysql做数据存储



# 目录 Contents

- ◆ 初识 ElasticSearch
- ◆ 安装 ElasticSearch
- ◆ ElasticSearch 核心概念
- ◆ 操作 ElasticSearch
- ◆ ElasticSearch JavaAPI

# 安装ElasticSearch

---

- 安装 ElasticSearch
- 安装 Kibana

# ■ 安装 Elasticsearch

请参考《ElasticSearch 安装.md》



# 目录 Contents

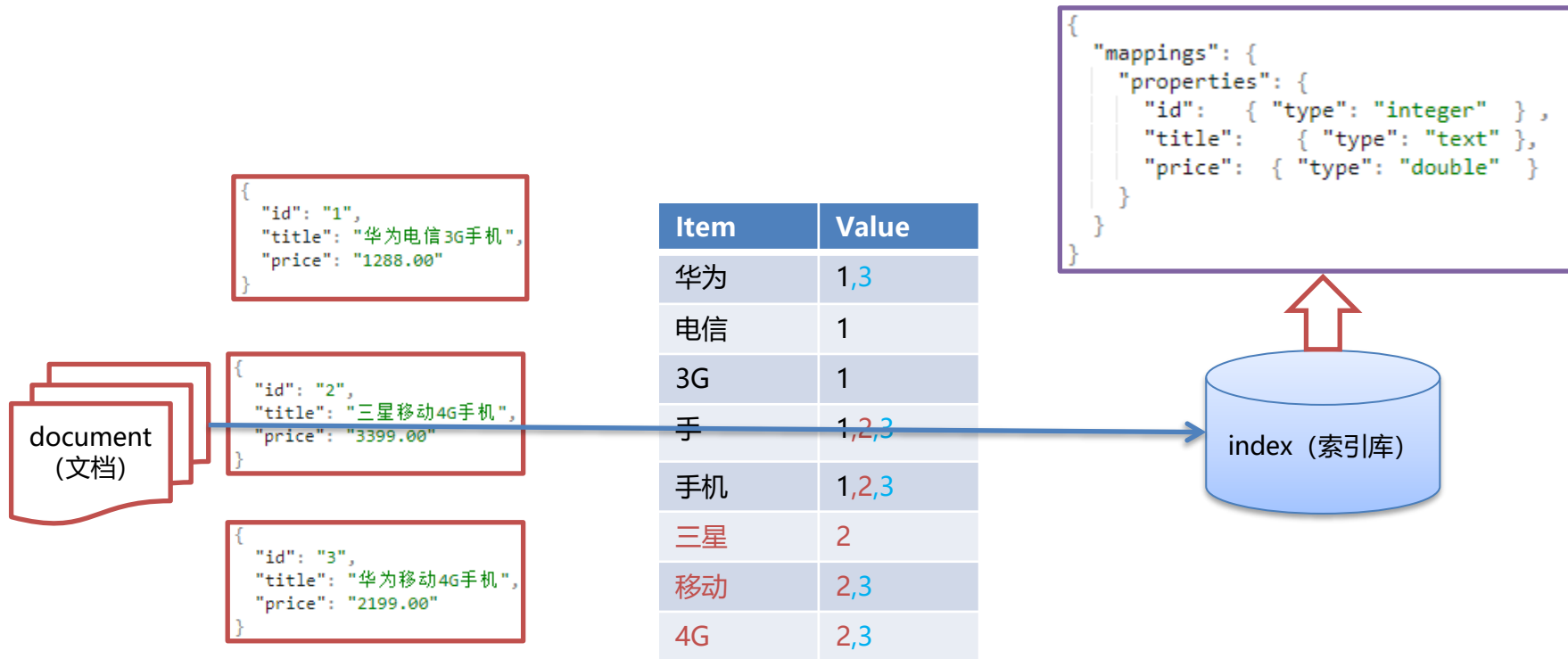
- ◆ 初识 ElasticSearch
- ◆ 安装 ElasticSearch
- ◆ ElasticSearch 核心概念
- ◆ 操作 ElasticSearch
- ◆ ElasticSearch JavaAPI

# ElasticSearch核心概念

---

- 索引 (index)
- 类型 (type)
- 映射 (mapping)
- 文档 (document)
- 倒排索引

# ElasticSearch 核心概念



- **索引 (index)**

ElasticSearch存储数据的地方，可以理解成关系型数据库中的数据库概念。

- **映射 (mapping)**

mapping定义了每个字段的类型、字段所使用的分词器等。相当于关系型数据库中的表结构。

- **文档 (document)**

Elasticsearch中的最小数据单元，常以json格式显示。一个document相当于关系型数据库中的一行数据。

- **倒排索引**

一个倒排索引由文档中所有不重复词的列表构成，对于其中每个词，对应一个包含它的文档id列表。

- **类型 (type)**

一种type就像一类表。如用户表、角色表等。在Elasticsearch7.X默认type为\_doc

- ES 5.x中一个index可以有多种type。
- ES 6.x中一个index只能有一种type。
- ES 7.x以后，将逐步移除type这个概念，现在的操作已经不再使用，默认\_doc





# 目录 Contents

- ◆ 初识 ElasticSearch
- ◆ 安装 ElasticSearch
- ◆ ElasticSearch 核心概念
- ◆ 操作 ElasticSearch
- ◆ ElasticSearch JavaAPI

# 操作 Elasticsearch

---

- RESTful风格介绍
- 基于 postman 操作
- 基于 kibana 操作
- 操作索引
- 操作映射
- 操作文档

## RESTful风格

- REST (Representational State Transfer) , 表述性状态转移, 是一组架构约束条件和原则。满足这些约束条件和原则的应用程序或设计就是RESTful。就是一种定义接口的规范。
- 基于HTTP。
- 可以使用XML格式定义或JSON格式定义。
- 每一个URI代表1种资源。
- 客户端使用GET、POST、PUT、DELETE 4个表示操作方式的动词对服务端资源进行操作：
  - GET: 用来获取资源
  - POST: 用来新建资源 (也可以用于更新资源)
  - PUT: 用来更新资源
  - DELETE: 用来删除资源

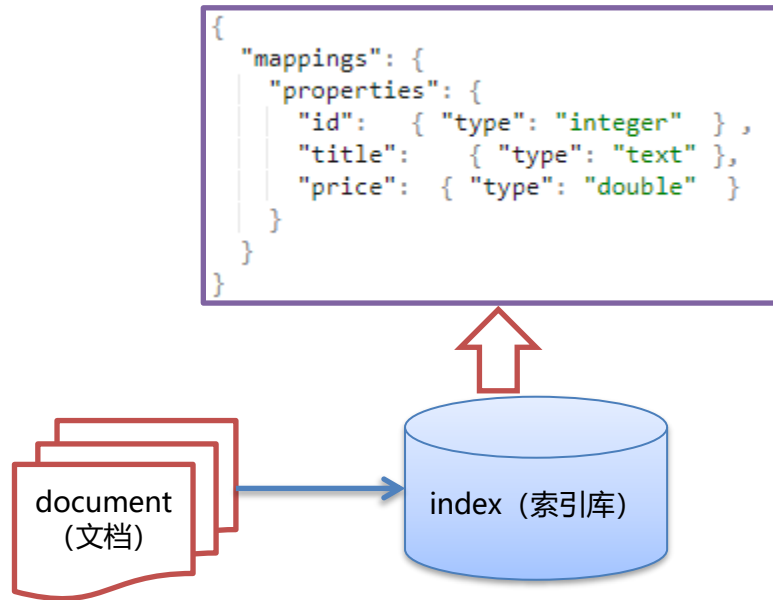
### 有一个/user资源

- get: /user/1 查询
- delete: /user/1 删除
- post: /user 添加
- update: /user 修改



## 操作索引

- 添加索引
- 查询索引
- 删除索引
- 关闭索引
- 打开索引





## 操作索引

- 添加索引

```
PUT http://ip:端口/索引名称
```

- 查询索引

```
GET http://ip:端口/索引名称 # 查询单个索引信息
```

```
GET http://ip:端口/索引名称1,索引名称2... # 查询多个索引信息
```

```
GET http://ip:端口/\_all # 查询所有索引信息
```

- 删除索引

```
DELETE http://ip:端口/索引名称
```

- 关闭索引

```
POST http://ip:端口/索引名称/\_close
```

- 打开索引

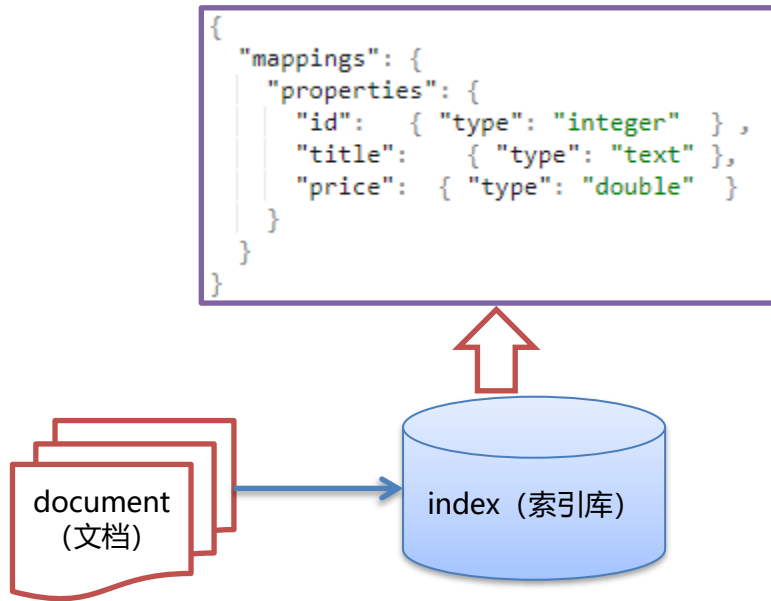
```
POST http://ip:端口/索引名称/\_open
```



## 操作映射

数据类型

- 简单数据类型
- 复杂数据类型



## 操作映射

华为手机

华为手机

### 简单数据类型

- 字符串
  - text: 会分词, 不支持聚合
  - keyword: 不会分词, 将全部
- 数值
- 布尔
  - boolean
- 二进制
  - binary
- 范围类型
  - integer\_range, float\_range, l
- 日期
  - date

long	带符号的64位整数, 其最小值为, 最大值为。 -263263-1
integer	带符号的32位整数, 其最小值为, 最大值为。 -231231-1
short	带符号的16位整数, 其最小值为-32,768, 最大值为32,767。
byte	带符号的8位整数, 其最小值为-128, 最大值为127。
double	双精度64位IEEE 754浮点数, 限制为有限值。
float	单精度32位IEEE 754浮点数, 限制为有限值。
half_float	半精度16位IEEE 754浮点数, 限制为有限值。
scaled_float	由a支持的有限浮点数long, 由固定double比例因子缩放。



## 操作映射

### 复杂数据类型

- 数组: []
- 对象: {}

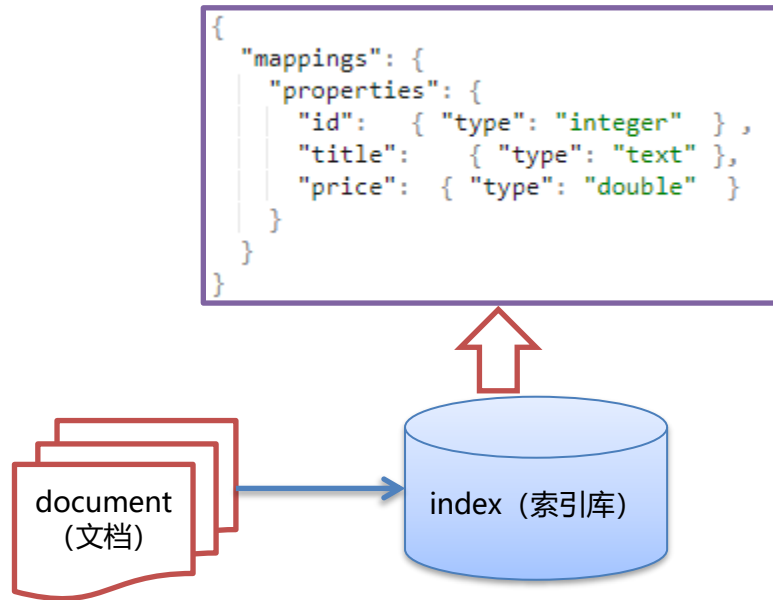
```
{
  "properties": {
    "name": { "type": "text" },
    "age": { "type": "integer" },
    "address": {
      "properties": {
        "province": { "type": "keyword" },
        "city": { "type": "keyword" },
        "district": { "type": "keyword" }
      }
    }
  }
}
```





## 操作映射

- 添加映射
- 查询映射
- 添加字段





## 操作映射

- 添加映射

```
# 添加映射
PUT /person/_mapping
{
  "properties":{
    "name":{
      "type":"text"
    },
    "age":{
      "type":"integer"
    }
  }
}
```

- 创建索引并添加映射

```
# 创建索引并添加映射
PUT /person
{
  "mappings":{
    "properties":{
      "name":{
        "type":"text"
      },
      "age":{
        "type":"integer"
      }
    }
  }
}
```

- 查询映射

```
# 查询映射
GET /person/_mapping
```

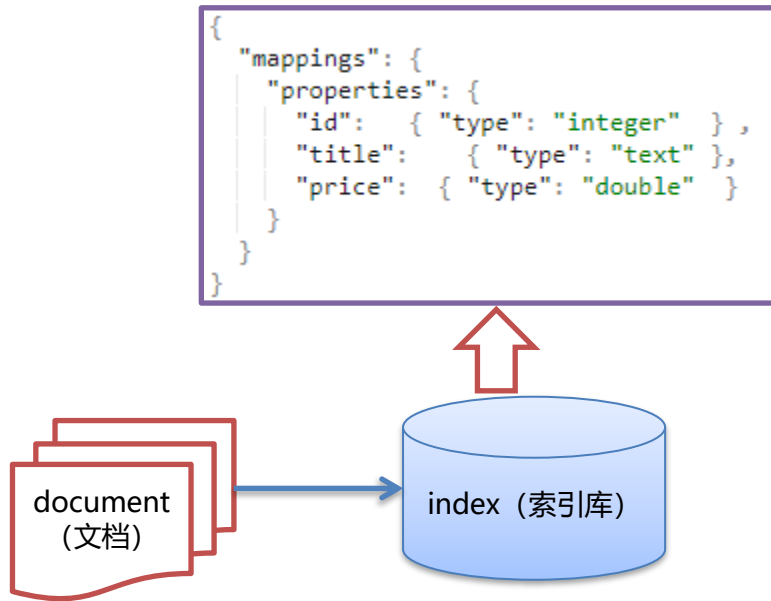
- 添加字段

```
# 添加字段
PUT /person/_mapping
{
  "properties":{
    "address":{
      "type":"text"
    }
  }
}
```



## 操作文档

- 添加文档
- 查询文档
- 修改文档
- 删除文档





## 操作文档

- 添加文档, 指定id

```
# 添加文档, 指定id
PUT person/_doc/1
{
  "name": "张三",
  "age": 20,
  "address": "北京市海淀区"
}
```

- 添加文档, 不指定id

```
POST person/_doc
{
  "name": "张三",
  "age": 20,
  "address": "北京市海淀区"
}
```

- 查询指定id的文档

```
# 查询指定id的文档
GET person/_doc/1
```

- 查询所有文档

```
# 查询所有文档
GET person/_search
```

- 删除指定id文档

```
# 删除指定id文档
DELETE person/_doc/1
```



## 分词器

- 分词器 (Analyzer) : 将一段文本, 按照一定逻辑, 分析成多个词语的一种工具  
如: 华为手机 --- > 华为、手、手机
- Elasticsearch 内置分词器
  - Standard Analyzer - 默认分词器, 按词切分, 小写处理
  - Simple Analyzer - 按照非字母切分(符号被过滤), 小写处理
  - Stop Analyzer - 小写处理, 停用词过滤(the,a,is)
  - Whitespace Analyzer - 按照空格切分, 不转小写
  - Keyword Analyzer - 不分词, 直接将输入当作输出
  - Patter Analyzer - 正则表达式, 默认\W+(非字符分割)
  - Language - 提供了30多种常见语言的分词器
- Elasticsearch 内置分词器对中文很不友好, 处理方式为: 一个字一个词



## IK分词器

- IKAnalyzer是一个开源的，基于java语言开发的轻量级的中文分词工具包
- 是一个基于Maven构建的项目
- 具有60万字/秒的高速处理能力
- 支持用户词典扩展定义
- 下载地址：<https://github.com/medcl/elasticsearch-analysis-ik/archive/v7.4.0.zip>



## 查询文档

- 词条查询: term
  - 词条查询不会分析查询条件, 只有当词条和查询字符串完全匹配时才匹配搜索
- 全文查询: match
  - 全文查询会分析查询条件, 先将查询条件进行分词, 然后查询, 求并集



## 查询文档

- 词条查询: term
  - 词条查询不会分析查询条件, 只有当词条和查询字符串完全匹配时才匹配搜索
- 全文查询: match
  - 全文查询会分析查询条件, 先将查询条件进行分词, 然后查询, 求并集

```
GET person/_search
{
  "query": {
    "term": {
      "name": {
        "value": "张三"
      }
    }
  }
}
```

```
GET person/_search
{
  "query": {
    "match": {
      "address": "北京"
    }
  }
}
```





# 目录 Contents

- ◆ 初识 ElasticSearch
- ◆ 安装 ElasticSearch
- ◆ ElasticSearch 核心概念
- ◆ 操作 ElasticSearch
- ◆ ElasticSearch JavaAPI

# ElasticSearch

---

# JavaAPI

- SpringBoot整合ES
- 操作索引
- 操作文档



## SpringBoot整合ES



### 案例：需求

使用SpringBoot整合ElasticSearch



### 案例：实现步骤

- ① 搭建SpringBoot工程
- ② 引入ElasticSearch相关坐标
- ③ 测试

```
<dependency>
  <groupId>org.elasticsearch.client</groupId>
  <artifactId>elasticsearch-rest-high-level-client</artifactId>
  <version>7.4.0</version>
</dependency>
<dependency>
  <groupId>org.elasticsearch.client</groupId>
  <artifactId>elasticsearch-rest-client</artifactId>
  <version>7.4.0</version>
</dependency>
<dependency>
  <groupId>org.elasticsearch</groupId>
  <artifactId>elasticsearch</artifactId>
  <version>7.4.0</version>
</dependency>
```



## 操作索引

- 添加索引
- 查询索引
- 删除索引
- 判断索引是否存在



## 操作文档

- 添加文档
- 修改文档
- 根据id查询文档
- 删除文档



传智播客旗下高端IT教育品牌