全部开发者教程

mono-repo)

7. go代码的检测工具

8. go中常见的错误

第22周 设计模式和单元测试

1. go最常用的设计模式 – 函数选项

2. 单例模式和懒加载

3. 测试金字塔

第23周 protoc插件开发、cobra命令行

1. protoc调试源码

2. protoc自定义gin插件

第24周 log日志包设计

日志源码

第25周 ast代码生成工具开发

错误码

第26周 三层代码结构

通用app项目启动

bobby · 更新于 2022-11-16

上一节 9. mapping 11. 分词的重要性 下一节

官方文档

Elasticsearch 中文本分析Analysis是把全文本转换成一系列的单词(term/token）的过程，也叫分词。文本分析是使用分析器 Analyzer 来实现的，Elasticsearch内置了分析器，用户也可以按照自己的需求自定义分析器。
为了提高搜索准确性，除了在数据写入时转换词条，匹配 Query 语句时候也需要用相同的分析器对查询语句进行分析。

Analyzer 的组成

Analyzer 由三部分组成：Character Filters、Tokenizer、Token Filters

Character Filters

Character Filters字符过滤器接收原始文本text的字符流，可以对原始文本增加、删除字段或者对字符做转换。一个**Analyzer 分析器可以有 0-n 个按顺序执行的字符过滤器**。
**

Tokenizer

Tokenizer 分词器接收Character Filters输出的字符流，将字符流分解成的那个的单词，并且输出单词流。例如空格分词器会将文本按照空格分解，将 “Quick brown fox!” 转换成 [Quick, brown, fox!]。分词器也负责记录每个单词的顺序和该单词在原始文本中的起始和结束偏移 offsets 。
一个Analyzer 分析器有且只有 1个分词器。

Token Filters

代码块

```
1  Token Filter单词过滤器接收分词器 Tokenizer 输出的单词流，可以对单词流中的单词做
```

一个Analyzer 分析器可以有 0-n 个按顺序执行的单词过滤器。

image.png

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，直接将输入当做输出
Patter Analyzer - 正则表达式，默认 \W+
Language - 提供了 30 多种常见语言的分词器

例子：The 2 QUICK Brown-Foxes jumped over the lazy dog's bone.

Standard Analyzer

- 默认分词器
- 按词分类
- 小写处理

代码块

```
#standard
GET _analyze
{
  "analyzer": "standard",
  "text": "The 2 QUICK Brown-Foxes jumped over the lazy dog's bone."
}
```

输出：
[the,2,quick,brown,foxes,a,jumped,over,the,lazy,dog's,bone]

Simple Analyzer

- 按照非字母切分，非字母则会被去除
- 小写处理

代码块

```
#simpe
GET _analyze
{
  "analyzer": "simple",
  "text": "The 2 QUICK Brown-Foxes jumped over the lazy dog's bone."
}
```

输出：
[the,quick,brown,foxes,jumped,over,the,lazy,dog,s,bone]

Stop Analyzer

- 小写处理
- 停用词过滤（the，a, is)

代码块

```
GET _analyze
{
  "analyzer": "stop",
```

意见反馈 收藏教程 标记书签

```
5       "text": "The 2 QUICK Brown-Foxes jumped over the lazy dog's bone."
    }
```

输出：

[quick,brown,foxes,jumped,over,lazy,dog,s,bone]

Whitespace Analyzer

- 按空格切分

代码块

```
#stop
GET _analyze
{
  "analyzer": "whitespace",
  "text": "The 2 QUICK Brown-Foxes jumped over the lazy dog's bone."
}
```

输出：

[The,2,QUICK,Brown-Foxes,jumped,over,the,lazy,dog's,bone.]

Keyword Analyzer

- 不分词，当成一整个 term 输出

代码块

```
#keyword
GET _analyze
{
  "analyzer": "keyword",
  "text": "The 2 QUICK Brown-Foxes jumped over the lazy dog's bone."
}
```

输出：

[The 2 QUICK Brown-Foxes jumped over the lazy dog's bone.]

Patter Analyzer

- 通过正则表达式进行分词
- 默认是 \W+(非字母进行分隔)

代码块

```
GET _analyze
{
  "analyzer": "pattern",
  "text": "The 2 QUICK Brown-Foxes jumped over the lazy dog's bone."
}
```

输出：

[the,2,quick,brown,foxes,jumped,over,the,lazy,dog,s,bone]

Language Analyzer

支持语言：arabic, armenian, basque, bengali, bulgarian, catalan, czech, dutch, english, finnish, french, portuguese,

意见反馈 收藏教程 标记书签

romanian, russian, sorani, spanish, swedish, turkish.

代码块

```
#english
GET _analyze
{
  "analyzer": "english",
  "text": "The 2 QUICK Brown-Foxes jumped over the lazy dog's bone."
}
```

输出：

[2,quick,brown,fox,jump,over,the,lazy,dog,bone]

中文分词要比英文分词难，英文都以空格分隔，中文理解通常需要上下文理解才能有正确的理解，比如
[苹果，不大好吃]和
[苹果，不大，好吃]，这两句意思就不一样。

常用的插件分词器

IK Analyzer - 对中文分词友好，支持远程词典热更新，有ik_smart 、ik_max_word 两种分析器
pinyin Analyzer - 可以对中文进行拼音分析，搜索时使用拼音即可搜索出来对应中文
ICU Analyzer - 提供了 Unicode 的支持，更好的支持亚洲语言
hanLP Analyzer - 基于NLP的中文分析器

9. mapping 上一节 下一节 11. 分词的重要性

我要提出意见反馈

企业服务 网站地图 网站首页 关于我们 联系我们 讲师招募 帮助中心 意见反馈 代码托管

意见反馈 收藏教程 标记书签