专栏：11：Elasticsearch 的使用

E - 爬虫技术　爬虫知识

> **用理工科思维看待这个世界**

系列爬虫专栏

崇尚的学习思维是：输入，输出平衡，且平衡点不断攀升。

曾经有大神告诫说：没事别瞎写文章；所以，很认真的写的是能力范围内的，看客要是看不懂，不是你的问题，问题在我，得持续输入，再输出。

今天的主题是：elasticsearch的使用

0：框架

序号	内容	说明
01	概念	–
02	安装及配置	–
03	使用	–
04	实战	–
05	总结及说明	–

1：概念

- Elasticsearch
  Elasticsearch 是一个实时分布式搜索和分析引擎。
- 特点
  分布式的实时文件存储，每个字段都被索引并可被搜索
  分布式的实时分析搜索引擎
  可以扩展到上百台服务器，处理PB级结构化或非结构化数据
- 索引 index (数据库)
  Elasticsearch 中存储数据的行为。
- 文档 type (表)
- 对比

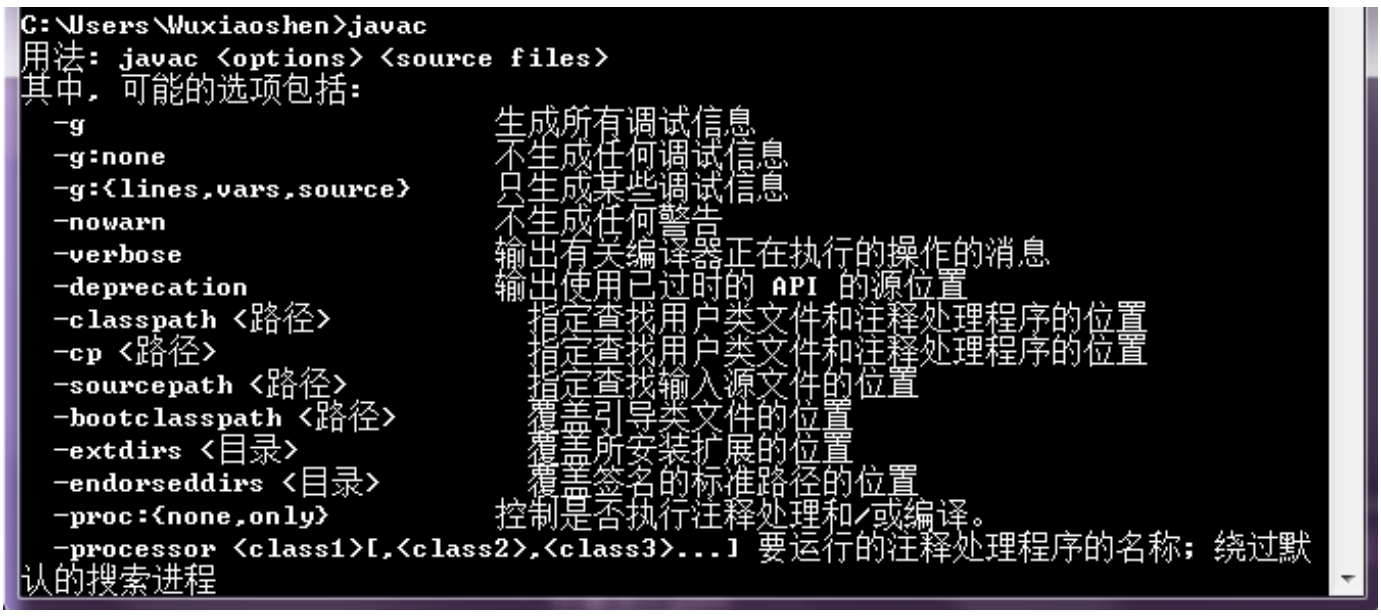
```
Relation DB   -> Databases -> tables -> rows      -> columns
Elasticsearch -> Indices   -> Types  -> Documents -> Fields
```

2：安装

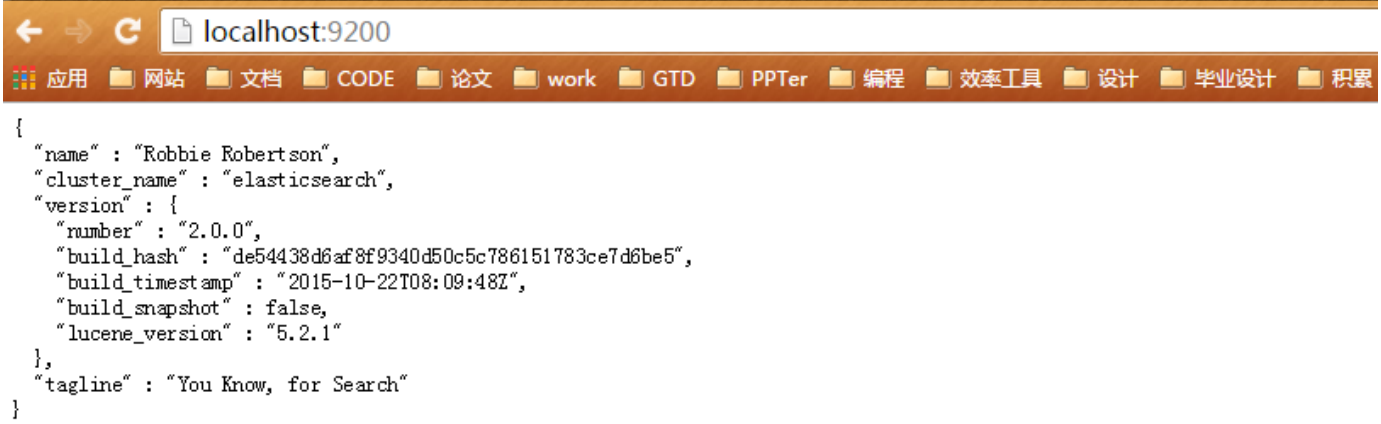
> windows 平台为例

- 下载Elasticsearch 官网 解压安装
- 下载JAVA官网 安装
- 配置JAVA环境变量
  - 环境变量
    - 新建系统变量JAVA_HOME 和 CLASSPATH
    - 变量名为：JAVA_HOME
    - 变量值：`C:\Program Files\Java\jdk1.8.0_65`
    - 变量名：CLASSPATH
    - >变量值：`%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar`
  - 选择“系统变量”中变量名为’path’ 的环境变量，添加JAVA绝对路径
    - 变量名：path
    - 变量值：`C:\Program Files\Java\jdk1.8.0_65\bin;C:\Program Files\Java\jre1.8.0_65`

验证是否配置正确JAVA环境：命令提示符下：javac



- 运行Elasticsearch
  目录下，命令提示符：./bin/elsticsearch
  浏览器中输入：`http://localhost:9200/`



- 安装插件：
  elasticsearch-head是一个elasticsearch的集群管理工具，它是完全由html5编写的独立网页程序
  目录下，命令提示符：`.\plugin install mobz/elasticsearch-head`
  浏览器中输入：`http://localhost:9200/_plugin/head/`

- 目录结构

目录	说明
bin	运行Elasticsearch 实例和插件管理所需的脚本
config	配置文件所在目录
lib	Elasticsearch所使用的库
data	存储ElasticSearch所使用的所有数据
logs	实例运行期间产生的事件和错误信息的文件
plugins	用于存储安装的插件
work	临时文件

更多操作官方网站

3：使用

在python中的使用为例
需要安装elasticsearch 库

```
# 向es中插入一条数据
from datetime import datetime
from elasticsearch import Elasticsearch
es = Elasticsearch()  # 创建连接

doc = {
    'author': 'kimchy',
    'text': 'Elasticsearch: cool. bonsai cool.',
    'timestamp': datetime.now(),
}                       # 文档
res = es.index(index="test-index", doc_type='tweet', id=1, body=doc) # 插入
数据
print(res['created'])
```

效果：


```
# 查询所有的test-index 下的数据
res = es.search(index="test-index", body={"query": {"match_all": {}}}) # 查
询数据，返回的是JSON格式的数据
print("Got %d Hits:" % res['hits']['total'])
for hit in res['hits']['hits']:
    print("%(timestamp)s %(author)s: %(text)s" % hit["_source"])

# output 上文插入的数据
Got 1 Hits:
2016-05-06T10:07:40.480466 kimchy: Elasticsearch: cool. bonsai cool.
```

管理工具上查询显示：同样是刚才那条数据

4：实战

> 核心代码参见专栏：009

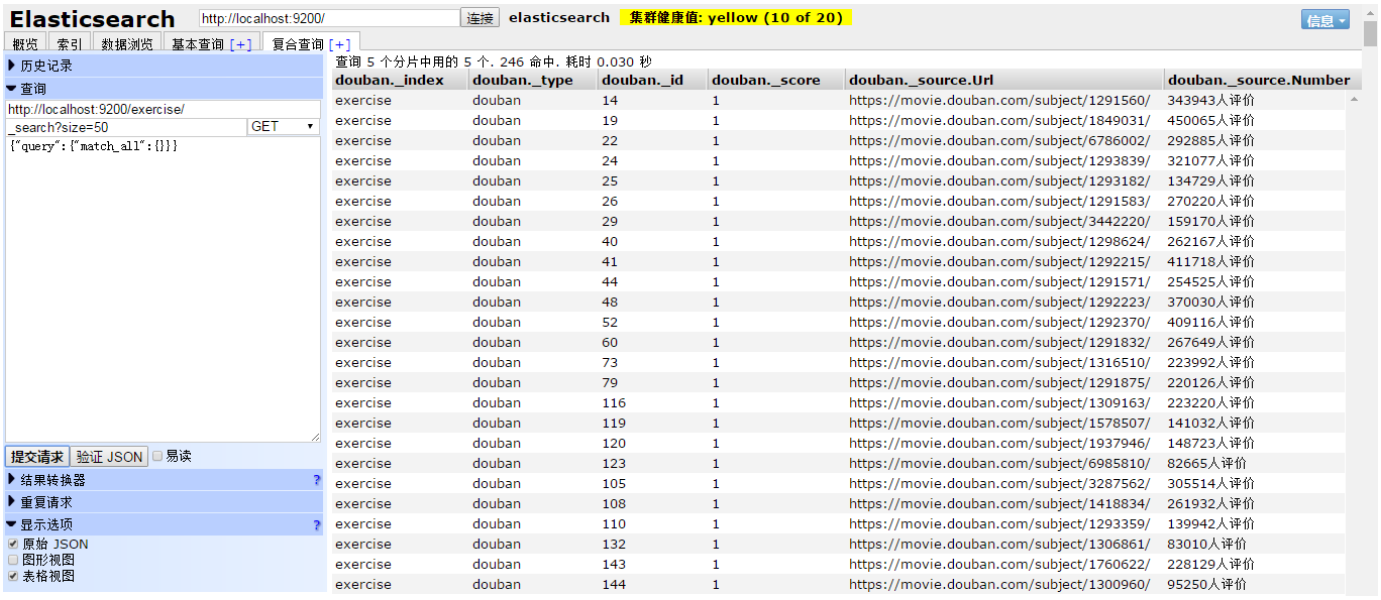
提供两种方式将抓取到的电影数据插入es中

```
# 第一种方式：content 是每部电影的数据：包括电影名，评分数，导演等
def save_to_es(self, content):
    global id
    data = json.dumps(content)
    url = "http://localhost:9200/exercise/douban/" + str(id)
    body = requests.post(url, data)
    id += 1
#......
if __name__ == "__main__":
    url = "https://movie.douban.com/top250?start=0&filter="
    Start = DouBanTop()
    urls = Start.urls()
    for one_url in urls:
        one_page_content = Start.get_content(one_url)
        all_data = Start.content_json(one_page_content)
        for one in all_data:
            Start.save_to_es(one)
```

```
    # 第二种使用elasticsearch库
    def save_to_es2(self, content):
        es = Elasticsearch()
        global id
        data = json.dumps(content)
        res = es.index(index="exercise", doc_type="douban", id = id, body = dat
    a)
        id +=1
    #............
    if __name__ == "__main__":
        url = "https://movie.douban.com/top250?start=0&filter="
        Start = DouBanTop()
        urls = Start.urls()
        for one_url in urls:
            one_page_content = Start.get_content(one_url)
            all_data = Start.content_json(one_page_content)
            for one in all_data:
                Start.save_to_es2(one)
```

效果显示：

查询效果显示：



完整版代码代码

5：参考及总结

参考文献：

1. elasticsearch文档
2. CSDN博客
3. 官方网站

Github:github
搭建了一个博客：博客

> IT初学者.