ElasticSearch使用说明

## ElasticSearch安装部署

注意使用2.3版本，否则得设置内存等，同时高版本的 ES，目前稳定版的 SpringBoot 1.5.13 的 SpringData 并不支持。

此处是在docker容器中安装部署。

### 下载elasticsearch

docker pull elasticsearch :2.3

### 下载elasticsearch-head

docker pull mobz/elasticsearch-head:5

### 设置elasticsearch配置文件

此处直接在docker容器服务器根目录建立/elasticsearch/es.yml。es.yml内容如下：

**#集群名称 所有节点要相同**

**cluster.name: "mangues\_es"**

**#本节点名称**

**node.name: master**

**#作为master节点**

**node.master: true**

**#是否存储数据**

**node.data: true**

**# head插件设置**

**http.cors.enabled: true**

**http.cors.allow-origin: "\*"**

**#设置可以访问的ip 这里全部设置通过**

**network.bind\_host: 0.0.0.0**

**#设置节点 访问的地址 设置master所在机器的ip**

**network.publish\_host: 192.168.105.81**

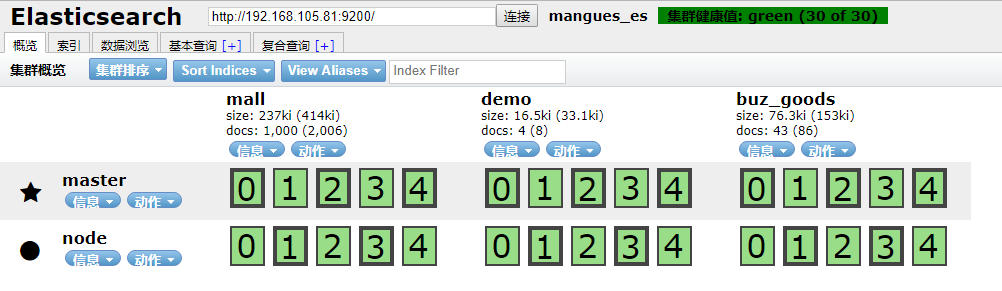
### 启动elasticsearch

docker run -d --name es -p 9200:9200 -p 9300:9300 -v /elasticsearch/es.yml:/usr/share/elasticsearch/config/elasticsearch.yml -v /elasticsearch/esdata1:/usr/share/elasticsearch/data elasticsearch:2.3

### 开启elasticsearch-head

docker run -p 9100:9100 mobz/elasticsearch-head:5

启动后，在浏览器打开地址<http://192.168.105.81:9100/>即可查看elasticsearch运行状态信息。



## ElasticSearch-IK中文分词器安装部署

### 进入elasticsearch容器

docker exec -it es /bin/bash

### 进入plugins目录

cd plugins

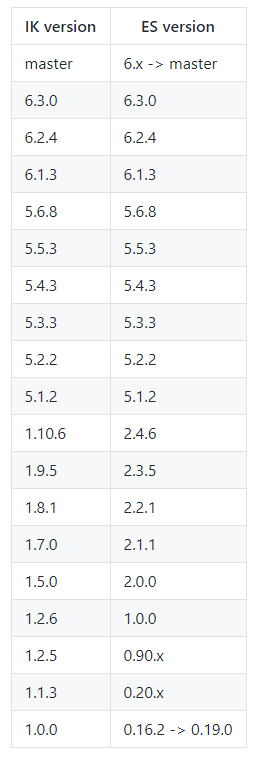
### 创建ik目录并进入

mkdir ik

cd ik

### 下载elasticsearch-analysis-ik插件

下载之前需要比对ik与elasticsearch的版本适配，具体如下图：



此处使用elasticsearch2.3，则对应下载ik版本为1.9.5

wget [https://github.com/medcl/elasticsearch-analysis-ik/releases/download/v1.9.5/elasticsearch-analysis-ik-1.9.5.zip](https://github.com/medcl/elasticsearch-analysis-ik/releases/download/v1.9.5/elasticsearch-analysis-ik-1.9.5.zip%20)

### 解压elasticsearch-analysis-ik安装包

unzip elasticsearch-analysis-ik-1.9.5.zip

### 删除elasticsearch-analysis-ik安装包

rm elasticsearch-analysis-ik-1.9.5.zip

然后重启elasticsearch服务即可。

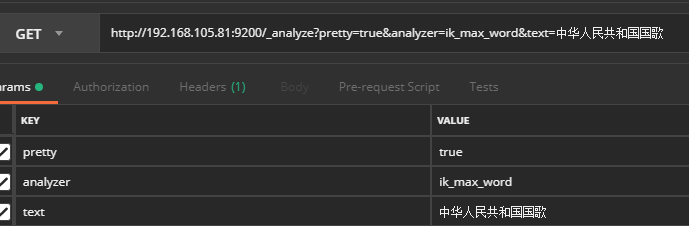
## ElasticSearch-IK中文分词说明

Ik具有两种分词模式，一种是最小粒度的ik\_max\_word ，另一种是最粗粒度的ik\_smart 。

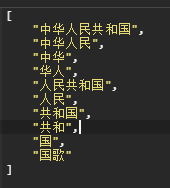
### 最小粒度模式ik\_max\_word

ik\_max\_word: 会将文本做最细粒度的拆分，比如会将“中华人民共和国国歌”拆分为“中华人民共和国,中华人民,中华,华人,人民共和国,人民,,共和国,共和,国,国歌”，会穷尽各种可能的组合。

测试结果，发送请求，如下图：



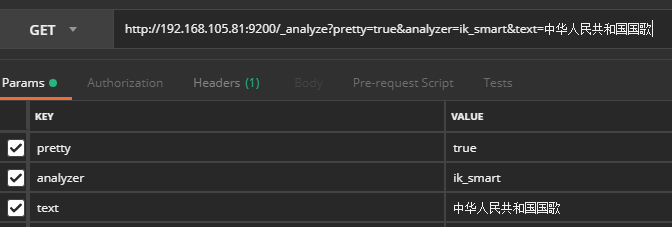
返回结果，如下图：



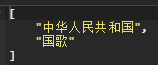
### 最粗粒度模式ik\_smart

ik\_smart: 会做最粗粒度的拆分，比如会将“中华人民共和国国歌”拆分为“中华人民共和国,国歌”。

测试结果，发送请求，如下图：



返回结果，如下图：



## Spring-data-elasticsearch开发详解

Elasticsearch概念解释：

索引（index）等同于数据库名称

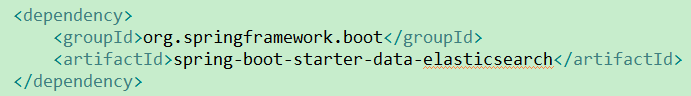
类型（type）等同于数据库表

文档（document）等同于表内一条数据

使用elasticsearch做全文检索，核心原则就是建库建表插入数据，再做相应的增删改查。

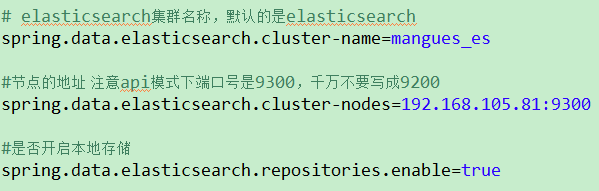
### pom.xml依赖

在pom.xml加入spring-data-elasticsearch依赖，如下图



### application.properties配置

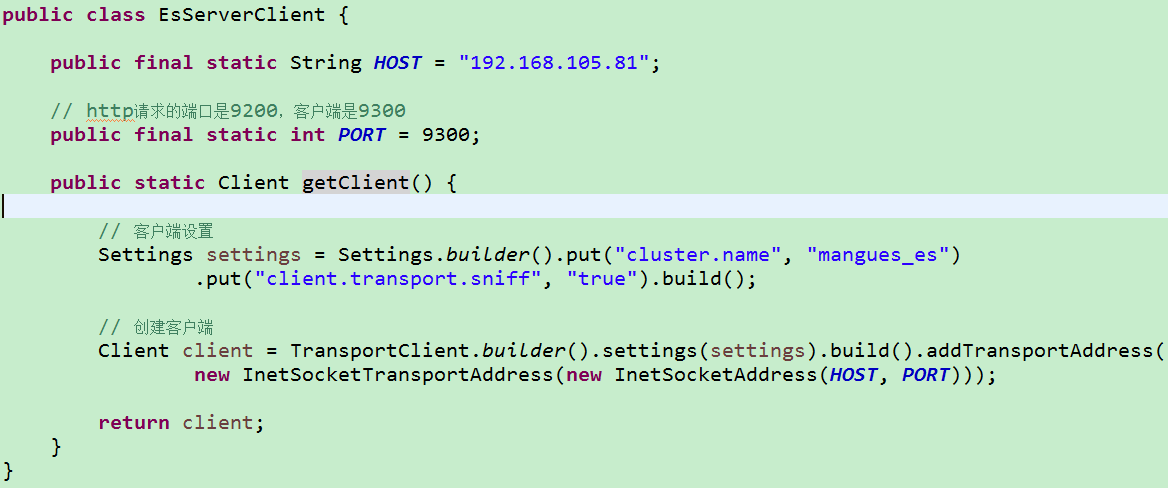
在application.properties文件中增加elasticsearch配置，如下图：



### 创建客户端连接

使用spring-data-elasticsearch进行操作之前，都需要先获取客户端连接，用完要关闭连接，释放资源，如下图：

创建连接：

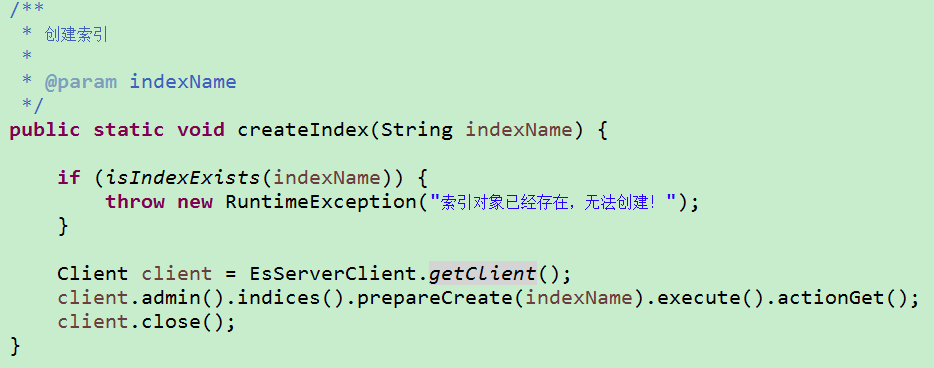


关闭连接：

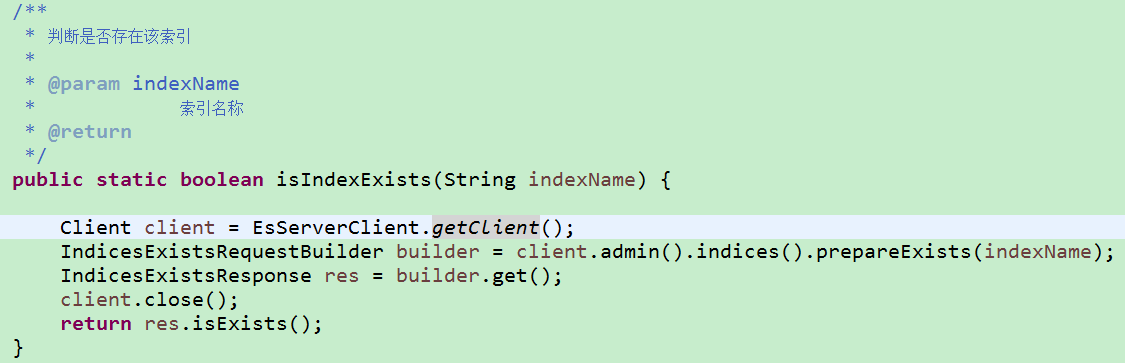


### 创建索引

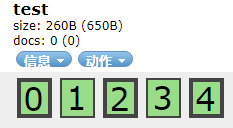
根据需要创建的索引名称，创建该索引：



上述代码片段，先进行索引是否存在的验证，isIndexExists方法：



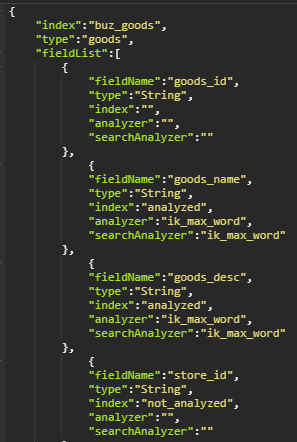
创建成功，刻在elasticsearch-head查看，如下图



### 创建类型及类型结构

此处为方便动态创建类型和类型结构，使用json参数构建类型结构的方式来实现。

请求参数：



index即在哪个索引下创建类型

type即创建类型的名称

fieldList为类型结构

fieldName为字段名称

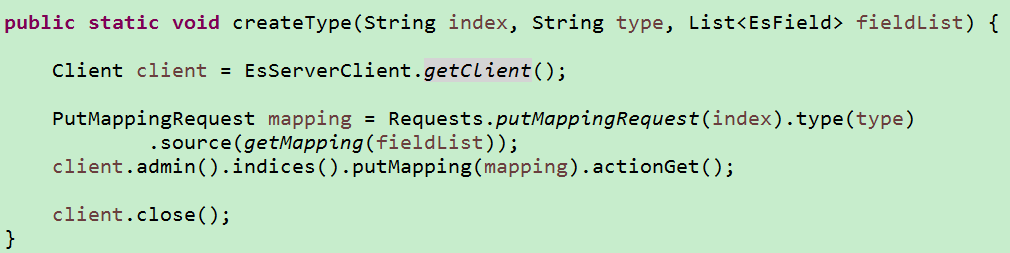
type为字段类型

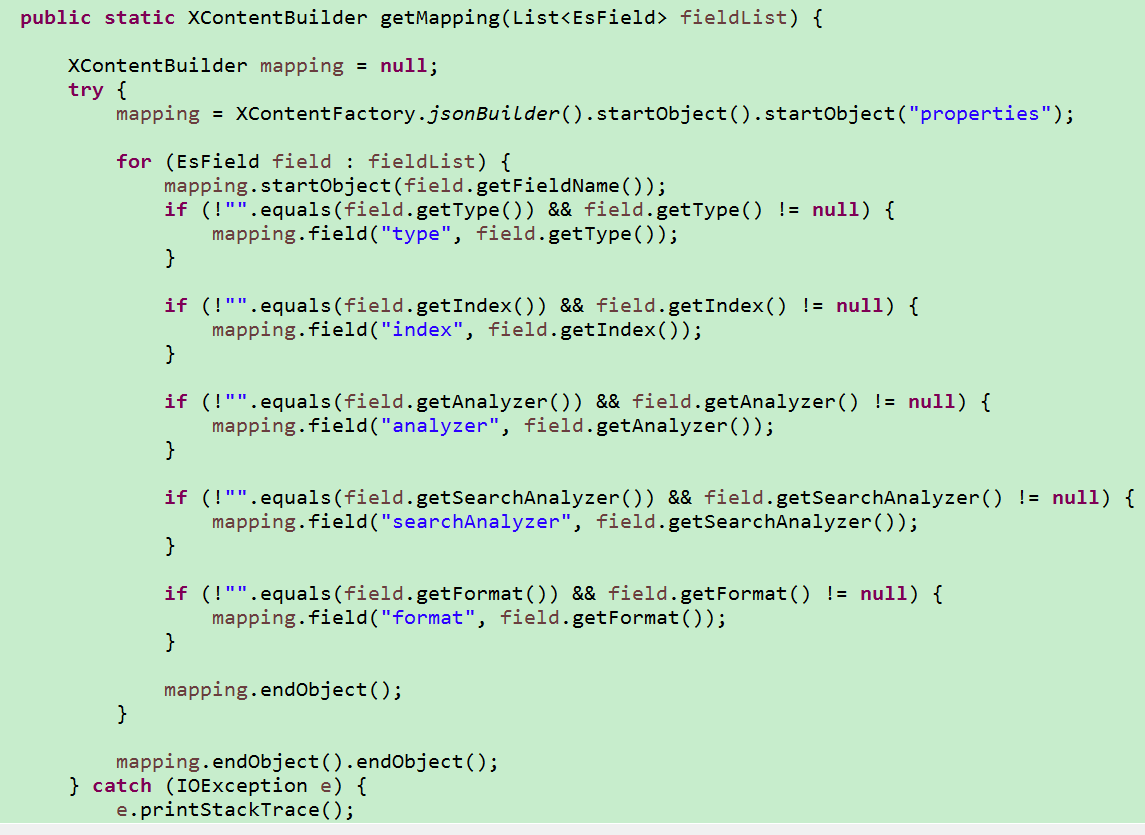
index为是否进行分词处理

analyzer为分词规则

searchAnalyzer为搜索时使用的分词规则

后台方法：

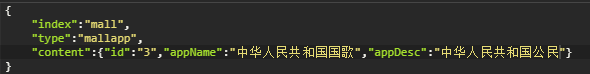




### 插入文档数据

现在索引，类型，都已创建完成，可以插入文档数据。

请求参数：

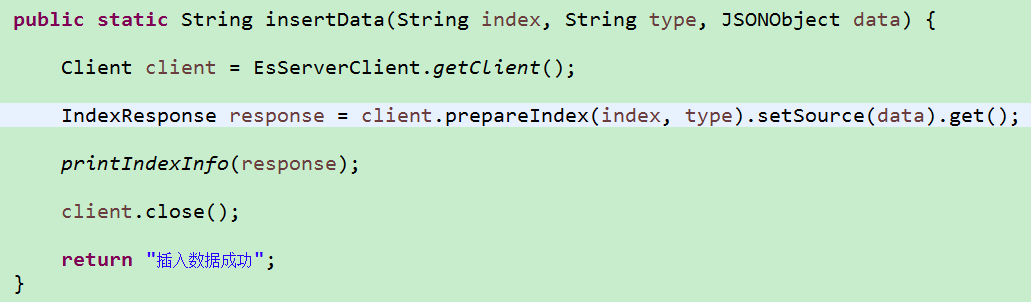


Index为插入数据的索引

Type为插入数据的类型

Content为插入的数据json

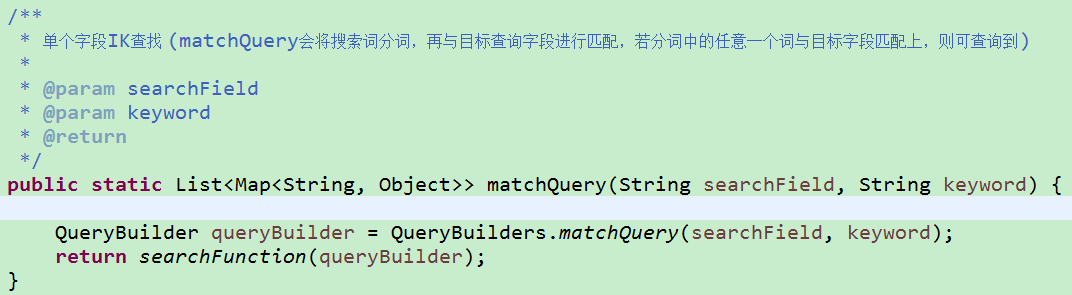
后台方法：



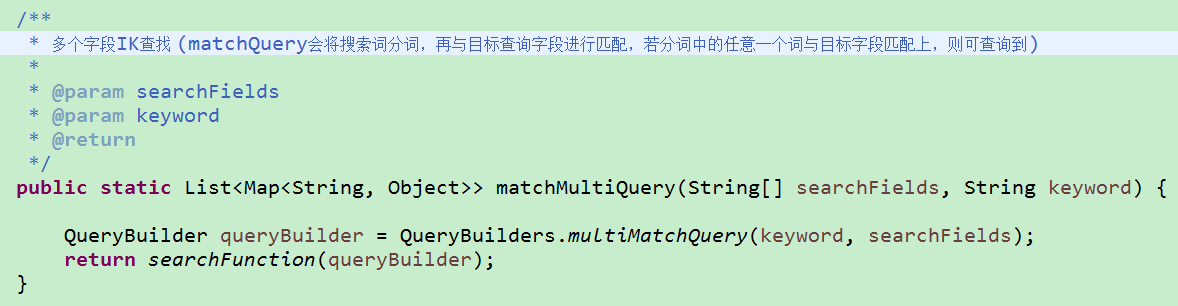
### 搜索文档数据

文档数据插入完毕，即可使用elasticsearch的检索功能。

根据单个字段和关键字进行搜索：



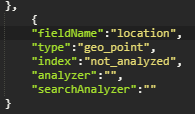
根据多个字段和关键字进行搜索：



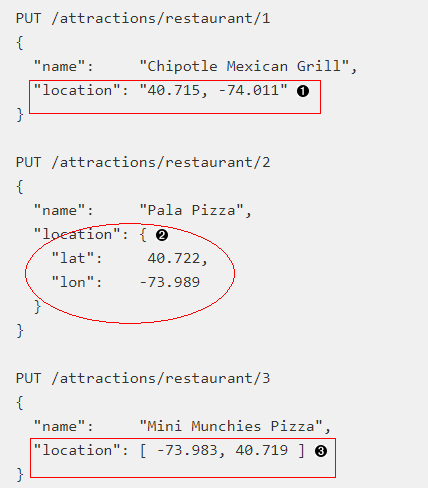
## 地理位置相关查询

### 经纬度坐标格式

作为经纬度字段类型必须为geo\_point，此处设置经纬度字段名为location，如下图：



该字段存放的数据可以是字符串，数组或对象，如下图：



字符串形式以半角逗号分割，如 "lat,lon" 。

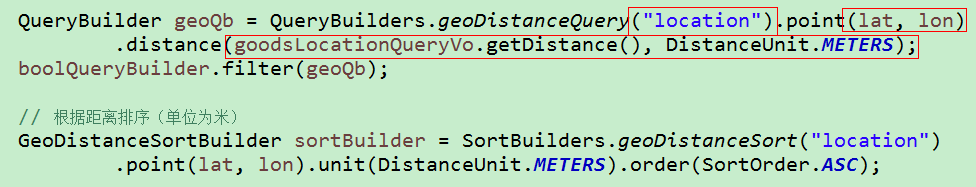
对象形式显式命名为 lat 和 lon 。

数组形式表示为 [lon,lat] 。

注意事项：地理坐标点用字符串形式表示时是纬度在前，经度在后（"latitude,longitude"），而数组形式表示时是经度在前，纬度在后（[longitude,latitude]）—顺序刚好相反。

### 目标点指定范围搜索

根据某个点的坐标位置，搜索以该点为圆心，指定距离的圆形范围内，所有符合条件的数据。



location经纬度字段名。

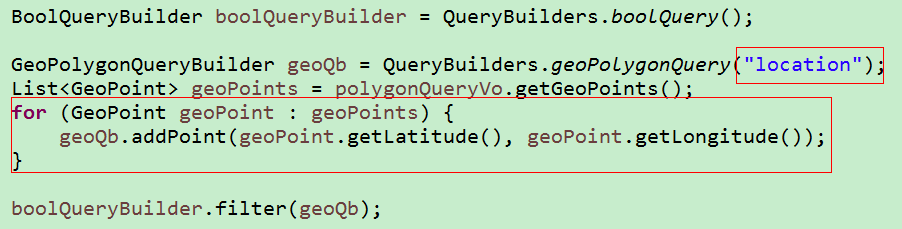
lat,lon指定点的纬度和经度。

distance指定搜索范围的距离

METERS距离单位，此处为“米”。

### 指定多边形地区搜索

多边形搜索，传入多边形各点坐标集合，查询位于坐标区域内符合条件的数据。

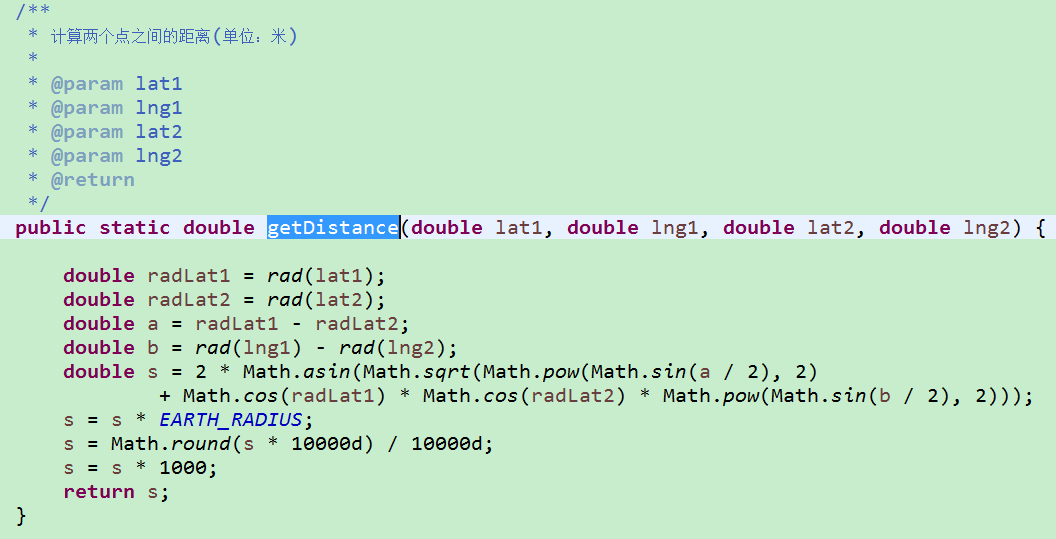


location经纬度字段名。

addPoint循环添加多边形每个坐标点。

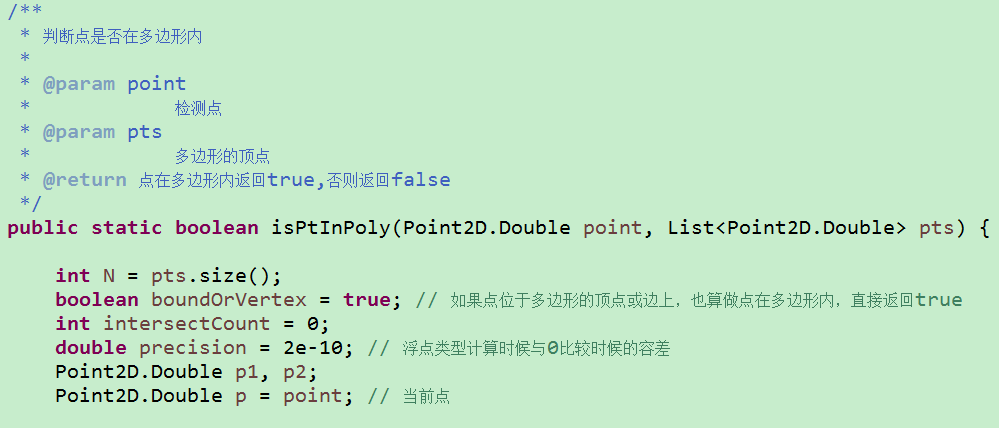
### 两点距离计算

根据两个点的坐标，通过地理计算公式，计算两点之间的距离。



### 计算目标点是否在指定多边形区域内

根据指定目标点，多边形坐标集合，计算该点是否在多边形内。



## 工程代码demo

## 工程代码打包上传至<https://lijingshan-1.ut.cn:8443/svn/SmartCloud/02document/技术预研/elasticsearch>该目录下，详细实现和其它操作见该工程。