

1.1 Solr服务的搭建

需要在linux系统下搭建solr服务。

- 1、需要安装jdk
- 2、安装tomcat

1.1.1 CentOS单机版安装：

第一步：安装jdk、安装tomcat

第二步：解压solr压缩包。

第三步：把dist/solr-4.10.3.war部署到tomcat下。

第四步：解压缩war包。启动tomcat解压。

第五步：需要把/root/solr-4.10.3/example/lib/ext目录下的所有的jar包添加到solr工程中。

第六步：创建solrhome。把/root/solr-4.10.3/example/solr文件夹复制一份作为solrhome。

第七步：告诉solr服务solrhome的位置。需要修改web.xml

```
WAR File can set the JNDI property here...
-->
<env-entry>
  <env-entry-name>solr/home</env-entry-name>
  <env-entry-value>/usr/local/solr/solrhome</env-entry-value>
  <env-entry-type>java.lang.String</env-entry-type>
</env-entry>

<!-- Any path (name) registered in solrconfig.xml will be sent to that fil
```

第八步：启动tomcat。

1.1.2 配置中文分析器、自定义业务域

分析器使用IKAnalyzer。

使用方法：

第一步：把IKAnalyzer依赖的jar包添加到solr工程中。把分析器使用的扩展词典添加到classpath中。

（在tomcat的webapps里面的solr工程的WEB-INF下创建classes文件夹，将ext_stopword.dic, mydict.dic, IKAnalyzer.cfg.xml拷贝到classes文件夹下）

第二步：需要自定义一个FieldType。Schema.xml中定义。可以在FieldType中指定中文分析器。

```
1 <fieldType name="text_ik" class="solr.TextField">
2   <analyzer class="org.wltea.analyzer.lucene.IKAnalyzer"/>
3 </fieldType>
```

第三步：自定义域。指定域的类型为自定义的FieldType。

```
1 <field name="item_title" type="text_ik" indexed="true" stored="true"/>
2 <field name="item_sell_point" type="text_ik" indexed="true" stored="true"/>
3 <field name="item_price" type="long" indexed="true" stored="true"/>
4 <field name="item_image" type="string" indexed="false" stored="true" />
5 <field name="item_category_name" type="string" indexed="true" stored="true" />
6 <field name="item_desc" type="text_ik" indexed="true" stored="false" />
7
8 <field name="item_keywords" type="text_ik" indexed="true" stored="false"
  multiValued="true"/>
9 <copyField source="item_title" dest="item_keywords"/>
10 <copyField source="item_sell_point" dest="item_keywords"/>
11 <copyField source="item_category_name" dest="item_keywords"/>
12 <copyField source="item_desc" dest="item_keywords"/>
```

由以上的field对应数据库中取出的数据属性

Sql语句:

```
1 SELECT
2   a.id,
3   a.title,
4   a.sell_point,
5   a.price,
6   a.image,
7   b.`name` category_name,
8   c.item_desc
9 FROM
10  tb_item a
11  LEFT JOIN tb_item_cat b ON a.cid = b.id
12  LEFT JOIN tb_item_desc c ON a.id = c.item_id
13 WHERE
14  a.`status` = 1
```

第四步：重新启动tomcat

1.2 索引库中导入导出数据

1.2.1 Solrj客户端的使用

```
1 public class SolrJTest {
2
3     @Test
4     public void testSolrJ() throws Exception {
5         //创建连接
6         SolrServer solrServer = new
7 HttpSolrServer("http://192.168.25.154:8080/solr");
8         //创建一个文档对象
9         SolrInputDocument document = new SolrInputDocument();
10        //添加域
11        document.addField("id", "solrtest01");
12        document.addField("item_title", "测试商品");
13        document.addField("item_sell_point", "卖点");
14        //添加到索引库
15        solrServer.add(document);
16        //提交
17        solrServer.commit();
18    }
19
20    @Test
21    public void testQuery() throws Exception {
22        //创建连接
23        SolrServer solrServer = new
24 HttpSolrServer("http://192.168.25.154:8080/solr");
25        //创建一个查询对象
26        SolrQuery query = new SolrQuery();
27        query.setQuery("*:~");
28        //执行查询
29        QueryResponse response = solrServer.query(query);
30        //取查询结果
31        SolrDocumentList solrDocumentList = response.getResults();
32        for (SolrDocument solrDocument : solrDocumentList) {
33            System.out.println(solrDocument.get("id"));
34            System.out.println(solrDocument.get("item_title"));
35            System.out.println(solrDocument.get("item_sell_point"));
36        }
37    }
38 }
```