# 1 启动solr

a.linux或者windows解压solr

b.启动命令：

自定义solr项目：java -Dsolr.solr.home=test-core -jar start.jar

用默认搜索项目：java -jar start.jar

|  |
| --- |
|  |

# 2 建立数据项目

a.在example目录下创建test-core文件夹；

b.将/solr下的solr.xml拷贝到test-core目录下；

c.在test-core下创建test目录，并且在test目录下创建conf和data目录；

|  |
| --- |
|  |

d.将example\solr\collection1\core.properties文件拷贝到test目录，修改name=test；

e.将example\solr\collection1\conf下的schema.xml、solrconfig.xml拷贝到example\test-core\test\conf下；

f.配置schema.xml文件：

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <schema name="example" version="1.7">  <field name="\_version\_" type="long" indexed="true" stored="true"/>  <field name="\_root\_" type="string" indexed="true" stored="false"/>  <field name="title" type="text\_ik" indexed="true" stored="true" required="true" multiValued="false" />  <!—复制域,不同字段检索结果取并集，检索复制域相当于检索所有复制域字段-->  <field name="copy\_item" type="text\_ik" indexed="true" stored="true" required="true" multiValued="false" />  <copyField source="title1" dest="copy\_item"/>  <copyField source="title2" dest="copy\_item"/>  <!—主键域-->  <uniqueKey>id</uniqueKey>    <fieldType name="string" class="solr.StrField" sortMissingLast="true" />  <fieldType name="long" class="solr.TrieLongField" precisionStep="0" positionIncrementGap="0"/>    <fieldtype name="binary" class="solr.BinaryField"/>  <fieldType name="text\_ik" class="solr.TextField">  <analyzer class="org.wltea.analyzer.lucene.IKAnalyzer"/>  </fieldType>  </schema> |

g 修改solrconfig.xml

|  |
| --- |
|  |

|  |
| --- |
|  |

# 3 配置分词和其他

a.将IKAnalyzer-2012-4x.jar拷贝到example\solr-webapp\webapp\WEB-INF\lib下；

b.在schema.xml文件中添加fieldType：

IKAnalyzer-2012-4x.jar

|  |
| --- |
| <fieldType name="text\_ik" class="solr.TextField">  <analyzer class="org.wltea.analyzer.lucene.IKAnalyzer"/>  </fieldType> |

c.重启solr，用analysis做分词测试

|  |
| --- |
|  |

# 4 导入数据到solr

a.配置solr搜索项目的集合字段

b.调用solr服务，导入数据

|  |
| --- |
| @Before  public void setUp() throws Exception {  // 在url中指定索索项目名称  String url = "http://localhost:8983/solr/test";  HttpSolrServer httpSolrServer = new HttpSolrServer(url); // 定义solr的server  httpSolrServer.setParser(new XMLResponseParser()); // 设置响应解析器  httpSolrServer.setMaxRetries(1); // 设置重试次数，推荐设置为1  httpSolrServer.setConnectionTimeout(500); // 建立连接的最长时间  this.httpSolrServer = httpSolrServer;  }  @Test  public void testImport() throws Exception {  // 查询商品数据  String url = "";  String jsonData = doGet(url);  List<OBJECT\_T\_MALL\_SKU> json\_to\_list = MyJsonUtil.json\_to\_list(jsonData, new OBJECT\_T\_MALL\_SKU());  this.httpSolrServer.deleteByQuery("\*:\*");  this.httpSolrServer.addBeans(json\_to\_list);  this.httpSolrServer.commit();  } |

c.删除solr数据，

<delete><query>\*:\*</query></delete>

this.httpSolrServer.deleteByQuery("\*:\*");

# 5 搜索项目工程

a.新建mvc项目

b.配置pom和spring容器中的solr客户端服务

|  |
| --- |
| <bean id ="solrServer" class="org.apache.solr.client.solrj.impl.HttpSolrServer">  <constructor-arg index="0" value="${url}" />  <property name="parser">  <bean class="org.apache.solr.client.solrj.impl.XMLResponseParser" />  </property>  <property name="maxRetries" value="1" />  <property name="connectionTimeout" value="${connectionTimeout}"/>  </bean> |

c.新建controller，返回为值可设置为jsp或者json均可

d.通过HttpSolrServer对solr服务进行检索，并处理检索结果

|  |
| --- |
| SolrQuery sq = new SolrQuery();  sq.setQuery("sku\_mch:联想高配高端");  QueryResponse query = null;  try {  query = solrServer.query(sq);  } catch (SolrServerException e) {  e.printStackTrace();  }  List<OBJECT\_T\_MALL\_SKU> beans = query.getBeans(OBJECT\_T\_MALL\_SKU.class); |

e.配置搜索关系

schema.xml中

|  |
| --- |
| <solrQueryParser defaultOperator="AND"/> |

# 6 solr端口号修改

1 修改搜索项目的配置文件

|  |
| --- |
|  |

2 修改默认端口号

|  |
| --- |
|  |

# 7访问url的问题

http://localhost:8983/solr/#/test 浏览器页面地址

http://localhost:8983/solr/test 访问solr搜索项地址

http://localhost:8983/solr/ solr服务地址

# 8 solr+zookeeper集群管理

1 zookeeper的配置

|  |
| --- |
| 1 在linux上准备多个zookeeper    2 配置zookeeper，加入Data目录，加入myid，myid中配置server编号，Zoo.cfg**中配置**  1 dataDir=/…  2 clientPort=218?  3 节点列表  server.1=ip:port:port  server.2=ip:port:port  …    3 启动所有的zookeeper节点  Sh zkServer.sh start  Sh zkServer.sh status |

2 solr项目的配置和启动

|  |
| --- |
| 1 解压solr复制war到tomcatwebapp下    2将solr中的ext目录lib目录下的jar拷贝到tomcat的solr/web-inf/lib下    3配置solrhome，选定目录如usr/local/solr/solrhome  修改tomcat中的solr项目的web.xml,将项目加载路径指向solrhome |

3 部署多个solr项目实例

|  |
| --- |
| 修改tomcat和solr的端口号，启动多个solr |

4 配置tomcat的java\_opts选项，在tomcat中配置zookeeper的集群ip

|  |
| --- |
| JAVA\_OPTS=”-DzkHost=ip:2181,ip:2182,ip:2183” |