# 前提

# 部署

使用tomcat运行solr

1. 获取war包：解压solr-4.10.3.tgz 进入solr-4.10.3\dist目录拿到war：solr-4.10.3.war放入tomcat中
2. 添加扩展包：位置在solr-4.10.3\example\lib\ext中的全部jar放入第一步的apache-tomcat-7.0.86\

webapps\solr\WEB-INF\lib下

1. 创建一个solrhome目录：位置在solr-4.10.3.tgz\solr-4.10.3\example\solr移到指定位置并改名solrhome
2. 配置solrhome位置：进入tomcat下的\webapps\solr\WEB-INF\web.xml打开进行修改：

<env-entry>

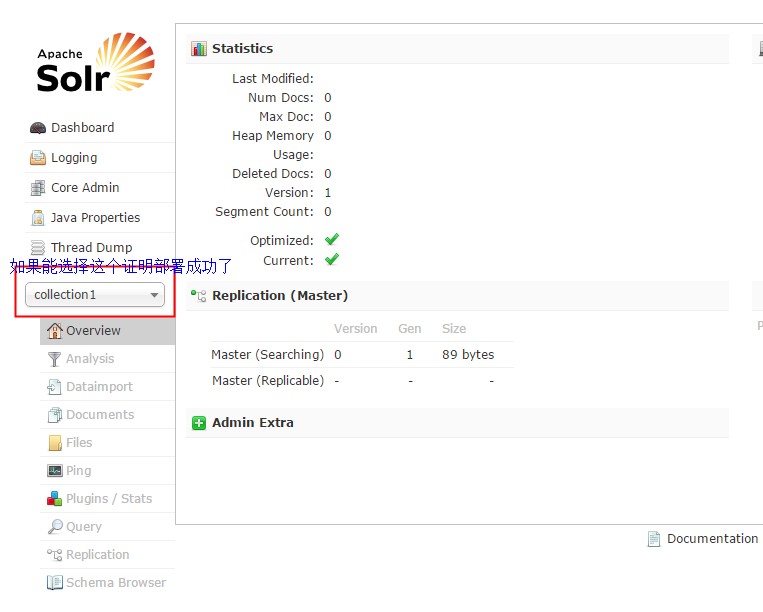
<env-entry-name>solr/home</env-entry-name>

<env-entry-value>I:\tomcat\tomcat-solr\apache-tomcat-7.0.86\solrhome</env-entry-value> //修改的位置

<env-entry-type>java.lang.String</env-entry-type>

</env-entry>

5、启动tomcat：访问http://localhost:8080/solr/



# 导入数据

## 第一步 分析

分析哪些表哪些字段需要导入sole库中

老师介绍的是

商品表中的字段：id、title(商品标题)、price(价格)、image(图片地址搜索结果页面展示)、cid(商品分类id)、商品状态(只需要正常的不需要下架的等状态)

## 第二步 配置中文分析器

solr本身不支持中文

部署中文分词器

1. 老师提供的分词器：IK Analyzer 2012FF\_hf1目录下有：IKAnalyzer2012FF\_u1.jar
2. 把IKAnalyzer2012FF\_u1.jar放入tomcat的solr中lib目录下
3. 把目录下IK Analyzer 2012FF\_hf1的mydict.dic(扩展词典)和ext\_stopword.dic(结构词典)IKAnalyzer.cfg.xml(配置文件)三个文件拷贝到tomcat的solr\WEB-INF\classes（此目录自己建）
4. 配置文件IKAnalyzer.cfg.xml：用来指定结构词典和扩展词典位置
5. 重要：定义分析器让solr找到：打开solrhome\collection1\conf\schema.xml拉到最后面配置下面内容：

参数：name：自己定义(不要和xml中id重复) class：solr的一种类型：必须是solr.TextField

<fieldType name="text\_ik" class="solr.TextField">

//这个类是是第2步jar包中的一个类：分析器

<analyzer class="org.wltea.analyzer.lucene.IKAnalyzer"/>

</fieldType>

## 第三步 配置业务域

在第二步中配置好中文分析器schema.xml最后面(也就是配置好的中文分析下面配置)

第一步 分析的有以下要定义：

1、商品Id 2、商品标题 3、商品卖点 4、商品价格 5、商品图片 6、分类名称

注意：创建对应的业务域。需要先制定中文分析器。

注意：Solr本身有id属性，就不用定义商品id了

配置业务域，type制定使用自定义的FieldType(中文分析器)

<!--商品标题 -->

<field name="item\_title" type="text\_ik" indexed="true" stored="true"/>

<!--商品卖点-->

<field name="item\_sell\_point" type="text\_ik" indexed="true" stored="true"/>

<!--商品价格：默认的long型没有使用中文分析器-->

<field name="item\_price" type="long" indexed="true" stored="true"/>

<!--商品图片：不需要中文分析器：就是要找的图片的完整地址-->

<field name="item\_image" type="string" indexed="false" stored="true" /><

<!--分类名称：就是一个完整的名称，如手机类，你不能拆分为手或机把-->

<field name="item\_category\_name" type="string" indexed="true" stored="true" />

<!-- 以下是复制域：当我们添加数据的时候会自动添加到dest参数中的名称下，对搜索的优化，(共享数据，都可以查出来)

<field name="item\_keywords" type="text\_ik" indexed="true" stored="false" multiValued="true"/>

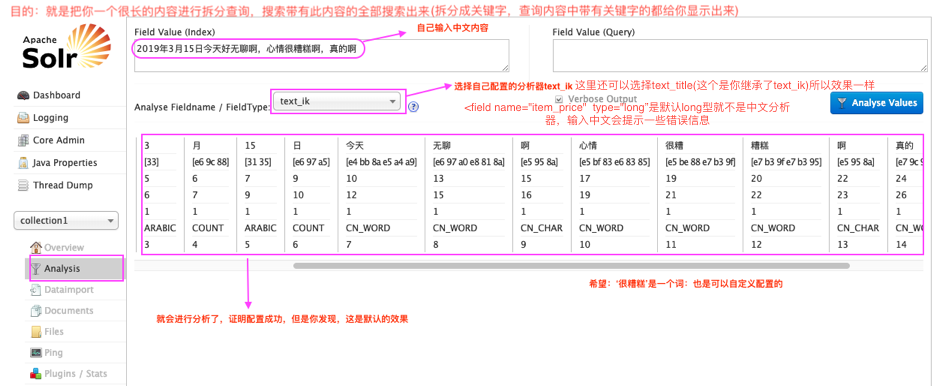
<copyField source="item\_title" dest="item\_keywords"/> <!-- 把标题自动复制到item\_keywords 下-->

<copyField source="item\_sell\_point" dest="item\_keywords"/>

<copyField source="item\_category\_name" dest="item\_keywords"/>

重启tomcat生效

## 第四步 查看分析器是否生效



## 第五步 导入数据

独立创建搜索服务工程：

e3-search(聚合工程) e3-search-interface(接口工程) e3-search-service(实现类)

1、需要连接数据库2不需要事务的配置3不需要redis缓存(solr自带)

导入数据连接数据库导入solr搜索服务库

分析：因为有两个表(部分字段)需要导入：需要自己写dao：sql语句进行关联要查的表

select \*(这里自己把需要字段全写出来(如果有冲突使用别名)) from tb\_item b left join tb\_cat b on a.cid=bid

独立建立一个pojo,用于接收自己关联表查询出来的数据：SearchItem

最简单的java代码solr服务器增删改操作步骤

导包

<!-- solr客户端 -->

<dependency>

<groupId>org.apache.solr</groupId>

<artifactId>solr-solrj</artifactId>

<version> 4.10.3</version>

</dependency>

代码如下

public class TestSolrJ {

//添加和更新(都一样只要键相同一致)即可数据到solr服务器中-操作--------------------------------------

@Test

public void addDocument() throws Exception {

//第一步 创建一个SolrServer对象，创建一个连接。参数solr服务的url

SolrServer solrServer = new HttpSolrServer("http://192.168.25.163:8080/solr/collection1");

//第二步 创建一个文档对象SolrInputDocument

SolrInputDocument document = new SolrInputDocument();

//第三步 向文档对象中添加域。文档中必须包含一个id域，所有的域的名称必须在schema.xml中定义。

document.addField("id", "doc01");

document.addField("item\_title", "测试商品01");

document.addField("item\_price", 1000);

//第四步 把文档写入索引库

solrServer.add(document);

//第五步 提交

solrServer.commit();

}

//删除操作-----------------

@Test

public void deleteDocument() throws Exception {

SolrServer solrServer = new HttpSolrServer("http://192.168.25.163:8080/solr/collection1");

//删除文档

//solrServer.deleteById("doc01");第一种方式

solrServer.deleteByQuery("id:doc01");第二种方式

//提交

solrServer.commit();

}

//查询操作-----------------

@Test

public void queryIndex() throws Exception {

//创建一个SolrServer对象。

SolrServer solrServer = new HttpSolrServer("http://192.168.25.163:8080/solr/collection1");

//创建一个SolrQuery对象。

SolrQuery query = new SolrQuery();

//设置查询条件。

//query.setQuery("\*:\*");第一种方式

query.set("q", "\*:\*"); 第二种方式

//执行查询，QueryResponse对象。

QueryResponse queryResponse = solrServer.query(query);

//取文档列表。取查询结果的总记录数 SolrDocumentList就是一个list集合

SolrDocumentList solrDocumentList = queryResponse.getResults();

System.out.println("查询结果总记录数：" + solrDocumentList.getNumFound());

//遍历文档列表，从取域的内容。查询结果是默认10条

for (SolrDocument solrDocument : solrDocumentList) {

System.out.println(solrDocument.get("id"));

System.out.println(solrDocument.get("item\_title"));

System.out.println(solrDocument.get("item\_sell\_point"));

System.out.println(solrDocument.get("item\_price"));

System.out.println(solrDocument.get("item\_image"));

System.out.println(solrDocument.get("item\_category\_name"));

}

}

//有条件的查询：如高亮

@Test

public void queryIndexFuza() throws Exception {

SolrServer solrServer = new HttpSolrServer("http://192.168.25.163:8080/solr/collection1");

//创建一个查询对象

SolrQuery query = new SolrQuery();

//查询条件

query.setQuery("手机");

query.setStart(0);

query.setRows(20);

query.set("df", "item\_title");

query.setHighlight(true);

query.addHighlightField("item\_title");

query.setHighlightSimplePre("<em>");

query.setHighlightSimplePost("</em>");

//执行查询

QueryResponse queryResponse = solrServer.query(query);

//取文档列表。取查询结果的总记录数

SolrDocumentList solrDocumentList = queryResponse.getResults();

System.out.println("查询结果总记录数：" + solrDocumentList.getNumFound());

//遍历文档列表，从取域的内容。

Map<String, Map<String, List<String>>> highlighting = queryResponse.getHighlighting();

for (SolrDocument solrDocument : solrDocumentList) {

System.out.println(solrDocument.get("id"));

//取高亮显示

List<String> list = highlighting.get(solrDocument.get("id")).get("item\_title");

String title = "";

if (list !=null && list.size() > 0 ) {

title = list.get(0);

} else {

title = (String) solrDocument.get("item\_title");

}

System.out.println(title);

System.out.println(solrDocument.get("item\_sell\_point"));

System.out.println(solrDocument.get("item\_price"));

System.out.println(solrDocument.get("item\_image"));

System.out.println(solrDocument.get("item\_category\_name"));

}

}

}

正常业务操作

配置web层：



导入数据：配置一个bean 让他单例不要new

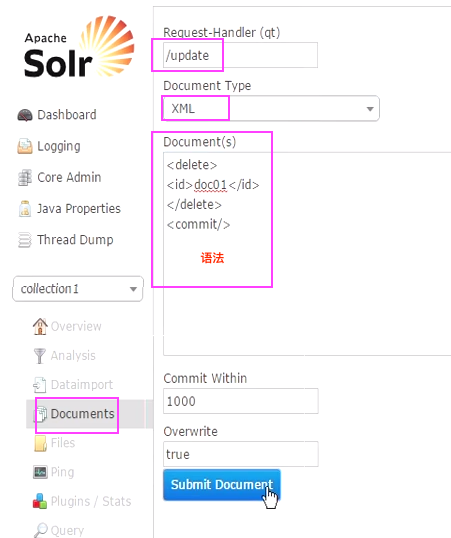
SolrServer solrServer = new HttpSolrServer("http://192.168.25.163:8080/solr/collection1");

## 第六步 查询操作

前台客户：web搜索页面：需要高亮、需要设置条件等比较麻烦

## Solr服务操作

删除操作数据操作



查询

比较麻烦：看32期07天12集内容