1.4、solrj的使用\_代码开始

类别 solr

# 1、环境搭建

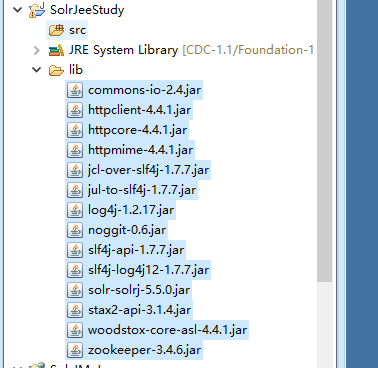
## 1：创建一个java工程。

## 2、：导入jar包。包括solrj的jar包

**1、solr-solrj-4.10.3.jar**

**2、solrj-lib文件夹下所有的jar包**

**3、还需要\solr-5.5.0\server\lib\ext目录下的所有的jar包。**



# 1、添加文档, SolrServer ,SolrInputDocument

## 1、代码如下，需要注意的是solr/core也就是说后面是索引库的名字

|  |
| --- |
| **public** **static** **void** **addDocument**() **throws** Exception {  //创建和solr服务器的连接  //参数就是solr服务器的url  SolrServer server = **new** HttpSolrServer("http://127.0.0.1:8080/solr/core");  //创建一个文档对象  SolrInputDocument document = **new** SolrInputDocument();  //向文档中添加域  //第一个参数是域的名称，必须在schema.xml中定义  //第二个参数是域的值  document.addField("id", "solr001");  document.addField("product\_name", "小米手机");  document.addField("product\_price", "2000");  //把文档添加到索引库  server.add(document);  server.commit();  } |

## 2、执行，成功

# 2、删除文档

## 1、根据id删除

|  |
| --- |
| //根据id删除  @Test  **public** **void** **deleteDocumentById**() **throws** Exception {  SolrServer server = **new** HttpSolrServer("http://localhost:8080/solr");  //直接调用删除的方法  server.deleteById("solr001");  server.commit();  } |

## 2、条件删除，下面这个不能随便用，一用就全删了

|  |
| --- |
| //根据查询删除文档  @Test  **public** **static** **void** **deleteDocumentByQuery**() **throws** Exception {  SolrServer server = **new** HttpSolrServer("http://127.0.0.1:8080/solr/core");  //调用删除方法  server.deleteByQuery("\*:\*");  server.commit();  } |

# 3、修改文档

## 1、solrj，没有提供更新的方法，本质就是先删除，再添加

# 4、查询文档

## 1、简单的查询

### 解释：SolrQuery ，QueryResponse，SolrDocumentList, SolrDocument,，默认返回10条

### 1、代码

|  |
| --- |
| //查询索引库  @Test  **public** **static** **void** **queryIndexSimple**() **throws** Exception {  SolrServer server = **new** HttpSolrServer("http://127.0.0.1:8080/solr/core");  //创建一个查询对象  SolrQuery query = **new** SolrQuery();  //添加查询条件.与下面的等价  //query.setQuery("\*:\*");  query.set("q", "\*:\*");  //执行查询  QueryResponse queryResponse = server.query(query);  //遍历结果  //取文档列表  SolrDocumentList results = queryResponse.getResults();  System.out.println("查询结果总数量：" + results.getNumFound());  **for** (SolrDocument solrDocument : results) {  System.out.println(solrDocument.get("id"));  System.out.println(solrDocument.get("product\_name"));  System.out.println(solrDocument.get("product\_price"));  System.out.println(solrDocument.get("product\_catalog\_name"));  System.out.println(solrDocument.get("product\_picture"));  }  } |

## 2、复杂查询

### 1、代码

|  |
| --- |
| //查询索引库，复杂查询  @Test  **public** **static** **void** **queryIndexFuza**() **throws** Exception {  SolrServer server = **new** HttpSolrServer("http://127.0.0.1:8080/solr/core");  //创建一个query对象  SolrQuery query = **new** SolrQuery();  //添加查询条件  query.setQuery("白雪公主");  //过滤条件  query.addFilterQuery("product\_catalog\_name:时尚卫浴", "product\_price:[0 TO 100]");  //排序条件  query.setSort("product\_price", ORDER.asc);  //分页条件  query.setStart(0);  query.setRows(15);  //设置返回结果中文档的域列表  query.setFields("id", "product\_name", "product\_price", "product\_catalog\_name", "product\_picture");  //设置默认搜索域  query.set("df", "product\_keywords");  //设置高亮  query.setHighlight(**true**);  //设置高亮显示的域  query.addHighlightField("product\_name");  //高亮前缀  query.setHighlightSimplePre("<em>");  //高亮后缀  query.setHighlightSimplePost("</em>");  //执行查询  QueryResponse queryResponse = server.query(query);  //取文档列表  SolrDocumentList documentList = queryResponse.getResults();  System.out.println("一共查询到："+ documentList.getNumFound());  **for** (SolrDocument solrDocument : documentList) {  System.out.println(solrDocument.get("id"));  //取高亮结果  Map<String, Map<String, List<String>>> highlighting = queryResponse.getHighlighting();  List<String> list = highlighting.get(solrDocument.get("id")).get("product\_name");  String productName = "";  **if** (**null** != list && list.size() > 0) {  productName = list.get(0);  } **else** {  productName = (String) solrDocument.get("product\_name");  }  System.out.println(productName);  System.out.println(solrDocument.get("product\_price"));  System.out.println(solrDocument.get("product\_catalog\_name"));  System.out.println(solrDocument.get("product\_picture"));    }  } |