**每日作业卷**

**Lucene&Solr第2天**

# 关卡1

## 训练案例1

### 训练描述

使用前一天的老师提供的课堂代码，实现以下功能：

要求：

1. 修改创建索引代码, 使用常用的Field类型
2. 实现修改索引
3. 实现删除索引
4. 运用Query子类查询索引库
5. 运用查询解析器查询索引库

### 操作步骤描述答案

1. 修改创建索引代码, 使用常用的Field类型

根据需求，选择使用常用的Field类型

// 图书ID，不用分词，不索引，要存储

doc.add(**new** StoredField("id", book.getId()));

// 图书名称,要分词，要索引，要存储

doc.add(**new** TextField("name", book.getName(), Store.***YES***));

// 图书价格，要分词，要索引，要存储

doc.add(**new** FloatField("price", book.getPrice(), Store.***YES***));

// 图书图片,不分词，不索引，要存储

doc.add(**new** StoredField("pic", book.getPic()));

// 图书描述，要分词，要索引，不存储

doc.add(**new** TextField("desc", book.getDesc(), Store.***NO***));

1. 实现修改索引

根据Term修改索引

**public** **class** UpdateIndex {

**private** IndexWriter indexWriter;

@Before

**public** **void** init() **throws** Exception {

// 创建IndexWriter

Analyzer analyzer = **new** IKAnalyzer();

Directory directory = FSDirectory.*open*(**new** File("C:/20161119/index"));

IndexWriterConfig config = **new** IndexWriterConfig(Version.***LUCENE\_4\_10\_3***, analyzer);

**this**.indexWriter = **new** IndexWriter(directory, config);

}

@Test

**public** **void** testUpdateIndex() **throws** Exception {

// 声明需要更新的数据

Document doc = **new** Document();

doc.add(**new** TextField("name", "Java 编程思想第四版", Store.***YES***));

// 执行更新

**this**.indexWriter.updateDocument(**new** Term("name", "java"), doc);

// 释放资源

**this**.indexWriter.close();

}

}

1. 实现删除索引

使用Term和删除所有两种方式实现

@Test

**public** **void** testDeleteIndex() **throws** Exception {

// 使用Term项(语汇单元)进行删除。

// this.indexWriter.deleteDocuments(new Term("name", "solr"));

// 删除全部，慎用

**this**.indexWriter.deleteAll();

// 释放资源

**this**.indexWriter.close();

}

1. 运用Query子类查询索引库

使用TermQuery、NumericRangeQuery和BooleanQuery三个子类查询

@Test

**public** **void** testTremQuery() **throws** Exception {

Query query = **new** TermQuery(**new** Term("desc", "java"));

**this**.doSearch(query);

}

@Test

**public** **void** testNumericRangeQuery() **throws** Exception {

Query query = NumericRangeQuery.*newFloatRange*("price", 55f, 66f, **true**, **true**);

**this**.doSearch(query);

}

@Test

**public** **void** testBooleanQuery() **throws** Exception {

Query query1 = **new** TermQuery(**new** Term("desc", "java"));

Query query2 = NumericRangeQuery.*newFloatRange*("price", 55f, 66f, **true**, **true**);

// 创建组合查询的Query对象

BooleanQuery booleanQuery = **new** BooleanQuery();

// 组合查询条件

// 第一个参数是设置组合的条件，第二个参数是设置组合的规则

booleanQuery.add(query1, Occur.***MUST***);

booleanQuery.add(query2, Occur.***MUST\_NOT***);

**this**.doSearch(booleanQuery);

}

1. 运用查询解析器查询索引库

使用设置单个默认域和设置多个默认域两种方式查询

@Test

**public** **void** testSearcherIndex() **throws** Exception {

Analyzer analyzer = **new** StandardAnalyzer();

QueryParser queryParser = **new** QueryParser("desc", analyzer);

Query query = queryParser.parse("lucene AND java");

doSearch(query);

}

@Test

**public** **void** testSearcherIndex2() **throws** Exception {

Analyzer analyzer = **new** StandardAnalyzer();

String[] fields = { "name", "desc" };

MultiFieldQueryParser queryParser = **new** MultiFieldQueryParser(fields, analyzer);

Query query = queryParser.parse("lucene");

System.***out***.println(query);

doSearch(query);

}

# 关卡2

## 训练案例1

### 训练描述

1. 安装并部署solr服务到Tomcat
2. 准备solrj开发环境，加入相关jar包
3. 使用solrJ完成对索引库增加和修改
4. 使用solrJ完成对索引库删除
5. 使用solrJ完成对索引库的查询

### 操作步骤描述答案

1. 安装并部署solr服务到Tomcat
2. 准备solrj开发环境，加入相关jar包

使用课堂资料完成

1. 使用solrJ完成对索引库增加和修改

**private** HttpSolrServer httpSolrServer;

@Before

**public** **void** init() {

String baseURL = "http://127.0.0.1:8081/solr/";

**this**.httpSolrServer = **new** HttpSolrServer(baseURL);

}

@Test

**public** **void** testSaveAndUpdate() **throws** Exception {

SolrInputDocument doc = **new** SolrInputDocument();

doc.addField("id", "c001");

doc.addField("title", "今天我们第一次使用solr，初级目标是15k");

**this**.httpSolrServer.add(doc);

**this**.httpSolrServer.commit();

}

1. 使用solrJ完成对索引库删除

@Test

**public** **void** testDelete() **throws** Exception {

// 根据唯一域（id） 删除

// this.httpSolrServer.deleteById("change.me");

// 根据条件删除（可以删除全部，慎用）

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

// 提交

**this**.httpSolrServer.commit();

}

1. 使用solrJ完成对索引库的查询

@Test

**public** **void** testQuery() **throws** Exception {

// 创建SolrQuery查询对象

SolrQuery query = **new** SolrQuery();

// 设置查询语句

query.setQuery("title:solr");

// 使用HttpSolrServer执行查询，返回response

QueryResponse response = **this**.httpSolrServer.query(query);

// 解析response，获取results

SolrDocumentList results = response.getResults();

// 使用results获取查询数据的总条数

System.***out***.println("查询到的数据总条数是：" + results.getNumFound());

// 遍历结果集results，打印数据

**for** (SolrDocument solrDocument : results) {

// 解析solrDocument

// 获取id

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

// 获取title

System.***out***.println(solrDocument.get("title").toString());

}

}

# 关卡3

## 训练案例1

### 训练描述

使用前一天的关卡3的数据，完成以下要求

要求：

1. 修改创建索引代码, 使用常用的Field类型

2. 实现修改索引

3. 实现删除索引

4. 运用Query子类查询索引库

5. 运用查询解析器查询索引库

6. 实现数据在solr中的增删改查

提示：保存到solr索引库的时候，不要保存pic和desc两个属性