写文章的时候,感觉比较难写的就是标题,有时候不知道起什么名字好,反正这里写的都是关于lucene的一些简单的实例,就随便起啦.

Lucene 其实很简单的,它最主要就是做两件事:建立索引和进行搜索

来看一些在lucene中使用的术语,这里并不打算作详细的介绍,只是点一下而已----因为这一个世界有一种好东西，叫搜索。

IndexWriter:lucene中最重要的的类之一，它主要是用来将文档加入索引，同时控制索引过程中的一些参数使用。

Analyzer:分析器,主要用于分析搜索引擎遇到的各种文本。常用的有StandardAnalyzer分析器,StopAnalyzer分析器,WhitespaceAnalyzer分析器等。

Directory:索引存放的位置;lucene提供了两种索引存放的位置，一种是磁盘，一种是内存。一般情况将索引放在磁盘上；相应地lucene提供了FSDirectory和RAMDirectory两个类。

Document:文档;Document相当于一个要进行索引的单元，任何可以想要被索引的文件都必须转化为Document对象才能进行索引。

Field：字段。

IndexSearcher:是lucene中最基本的检索工具，所有的检索都会用到IndexSearcher工具;

Query:查询，lucene中支持模糊查询，语义查询，短语查询，组合查询等等,如有TermQuery,BooleanQuery,RangeQuery,WildcardQuery等一些类。

QueryParser: 是一个解析用户输入的工具，可以通过扫描用户输入的字符串，生成Query对象。

Hits:在搜索完成之后，需要把搜索结果返回并显示给用户，只有这样才算是完成搜索的目的。在lucene中，搜索的结果的集合是用Hits类的实例来表示的。

上面作了一大堆名词解释，下面就看几个简单的实例吧:

1、简单的的StandardAnalyzer测试例子

Java代码

package lighter.javaeye.com;

import java.io.IOException;

import java.io.StringReader;

import org.apache.lucene.analysis.Analyzer;

import org.apache.lucene.analysis.Token;

import org.apache.lucene.analysis.TokenStream;

import org.apache.lucene.analysis.standard.StandardAnalyzer;

public class StandardAnalyzerTest

{

//构造函数，

public StandardAnalyzerTest()

{

}

public static void main(String[] args)

{

//生成一个StandardAnalyzer对象

Analyzer aAnalyzer = new StandardAnalyzer();

//测试字符串

StringReader sr = new StringReader("lighter javaeye com is the are on");

//生成TokenStream对象

TokenStream ts = aAnalyzer.tokenStream("name", sr);

try {

int i=0;

Token t = ts.next();

while(t!=null)

{

//辅助输出时显示行号

i++;

//输出处理后的字符

System.out.println("第"+i+"行:"+t.termText());

//取得下一个字符

t=ts.next();

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

显示结果：

引用

第1行:lighter

第2行:javaeye

第3行:com

提示一下：

StandardAnalyzer是lucene中内置的"标准分析器",可以做如下功能:

1、对原有句子按照空格进行了分词

2、所有的大写字母都可以能转换为小写的字母

3、可以去掉一些没有用处的单词，例如"is","the","are"等单词，也删除了所有的标点

查看一下结果与"new StringReader("lighter javaeye com is the are on")"作一个比较就清楚明了。

这里不对其API进行解释了，具体见lucene的官方文档。需要注意一点，这里的代码使用的是lucene2的API，与1.43版有一些明显的差别。

2、看另一个实例,简单地建立索引，进行搜索

Java代码

package lighter.javaeye.com;

import org.apache.lucene.analysis.standard.StandardAnalyzer;

import org.apache.lucene.document.Document;

import org.apache.lucene.document.Field;

import org.apache.lucene.index.IndexWriter;

import org.apache.lucene.queryParser.QueryParser;

import org.apache.lucene.search.Hits;

import org.apache.lucene.search.IndexSearcher;

import org.apache.lucene.search.Query;

import org.apache.lucene.store.FSDirectory;

public class FSDirectoryTest {

//建立索引的路径

public static final String path = "c:\\index2";

public static void main(String[] args) throws Exception {

Document doc1 = new Document();

doc1.add( new Field("name", "lighter javaeye com",Field.Store.YES,Field.Index.TOKENIZED));

Document doc2 = new Document();

doc2.add(new Field("name", "lighter blog",Field.Store.YES,Field.Index.TOKENIZED));

IndexWriter writer = new IndexWriter(FSDirectory.getDirectory(path, true), new StandardAnalyzer(), true);

writer.setMaxFieldLength(3);

writer.addDocument(doc1);

writer.setMaxFieldLength(3);

writer.addDocument(doc2);

writer.close();

IndexSearcher searcher = new IndexSearcher(path);

Hits hits = null;

Query query = null;

QueryParser qp = new QueryParser("name",new StandardAnalyzer());

query = qp.parse("lighter");

hits = searcher.search(query);

System.out.println("查找\"lighter\" 共" + hits.length() + "个结果");

query = qp.parse("javaeye");

hits = searcher.search(query);

System.out.println("查找\"javaeye\" 共" + hits.length() + "个结果");

}

}

运行结果：

Java代码

查找"lighter" 共2个结果

查找"javaeye" 共1个结果