# 环境搭建

<dependencies>  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.12</version>  
 </dependency>  
 <dependency>  
 <groupId>org.apache.lucene</groupId>  
 <artifactId>lucene-core</artifactId>  
 <version>5.5.5</version>  
 </dependency>  
 <dependency>  
 <groupId>org.apache.lucene</groupId>  
 <artifactId>lucene-analyzers-common</artifactId>  
 <version>5.5.5</version>  
 </dependency>  
 <dependency>  
 <groupId>org.apache.lucene</groupId>  
 <artifactId>lucene-queryparser</artifactId>  
 <version>5.5.5</version>  
 </dependency>  
 <!-- 高亮 -->  
 <dependency>  
 <groupId>org.apache.lucene</groupId>  
 <artifactId>lucene-highlighter</artifactId>  
 <version>5.5.5</version>  
 </dependency>  
 <!-- 中文分词器 SmartChineseAnalyzer -->  
 <dependency>  
 <groupId>org.apache.lucene</groupId>  
 <artifactId>lucene-analyzers-smartcn</artifactId>  
 <version>5.5.5</version>  
 </dependency>  
 <!-- 文件操作jar包 -->  
 <dependency>  
 <groupId>commons-io</groupId>  
 <artifactId>commons-io</artifactId>  
 <version>2.4</version>  
 </dependency>  
</dependencies>

测试类：

**import org.apache.lucene.analysis.**Analyzer;  
**import org.apache.lucene.analysis.cn.smart.SmartChineseAnalyzer**;  
**import org.apache.lucene.document.Document**;  
**import org.apache.lucene.document.Field**;  
**import org.apache.lucene.document.StringField**;  
**import org.apache.lucene.document.TextField**;  
**import org.apache.lucene.index.\***;  
**import org.apache.lucene.queryparser.classic.ParseException**;  
**import org.apache.lucene.queryparser.classic.QueryParser**;  
**import org.apache.lucene.search.IndexSearcher**;  
**import org.apache.lucene.search.**Query;  
**import org.apache.lucene.search.ScoreDoc**;  
**import org.apache.lucene.search.TopDocs**;  
**import org.apache.lucene.store.**FSDirectory;  
**import org.junit.Test**;  
  
**import java.io.File**;  
**import java.io.IOException**;  
**import java.nio.file.Path**;  
  
**public class LuceneTest {  
 String**[] **ids = {"1"**, **"2"**, **"3"**, **"4"**, **"5"**, **"6"}**;  
  
 **String**[] **names = {"zs"**, **"ls"**, **"ww"**, **"hl"**, **"wq"**, **"bb"}**;  
  
 **String**[] **emails = {"zs@qq.com"**, **"zs@baidu.com"**, **"zs@126.com"**, **"zs@sina.com"**, **"zs@163.com"**, **"zs@google.com"}**;  
  
 **String**[] **contents = {  
 "She had been shopping with her Mom in Wal-Mart. She must have been 6 years old, this beautiful brown haired, freckle-faced image of innocence. It was pouring outside. The kind of rain that gushes over the top of rain gutters, so much in a hurry to hit the Earth, it has no time to flow down the spout."**,  
 **"We all stood there under the awning and just inside the door of the Wal-Mart. We all waited, some patiently, others irritated, because nature messed up their hurried day. I am always mesmerized by rainfall. I get lost in the sound and sight of the heavens washing away the dirt and dust of the world. Memories of running, splashing so carefree as a child come pouring in as a welcome reprieve from the worries of my day."**,  
 **"Her voice was so sweet as it broke the hypnotic trance we were all caught in, Mom, let's run through the rain. she said."**,  
 **"The entire crowd stopped dead silent. I swear you couldn't hear anything but the rain. We all stood silently. No one came or left in the next few minutes. Mom paused and thought for a moment about what she would say."**,  
 **"Now some would laugh it off and scold her for being silly. Some might even ignore what was said. But this was a moment of affirmation in a young child's life. Time when innocent trust can be nurtured so that it will bloom into faith."**,  
 **"To everything there is a season and a time to every purpose under heaven. I hope you still take the time to "}**;  
 **Path path = new** File(**"E:**\\**tmp**\\**lucene"**).toPath();  
 **//索引存放目录** FSDirectory **directory =** FSDirectory.open(**path**);  
 **//也可以存放到内存  
 //Directory directory = new RAMDirectory();  
 //2.创建分词器** Analyzer **analyzer = new** SmartChineseAnalyzer();  
 **//3.创建IndexWriterConfig  
 IndexWriterConfig iwc = new** IndexWriterConfig(**analyzer**);  
  
 **public** LuceneTest() **throws IOException {  
 }  
  
  
 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 增加索引(Create) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
 @Test  
 public void** createIndex() **{  
  
 //4.创建IndexWriter  
 IndexWriter iw = null**;  
 **try {  
 // 创建writer  
 iw = new** IndexWriter(**directory**, **iwc**);  
 **for** (**int i = 0**; **i < ids**.**length**; **i++**) **{  
 Document doc = new** Document();  
 **doc**.add(**new** StringField(**"id"**, **ids**[**i**], **Field**.Store.***YES***));  
 **doc**.add(**new** StringField(**"name"**, **names**[**i**], **Field**.Store.***YES***));  
 **Field field = new** TextField(**"email"**, **emails**[**i**], **Field**.Store.***YES***);  
 **doc**.add(**field**);  
  
 **// 加权操作。qq邮箱2.0 新浪有限1.5 其他默认1.0 谷歌0.5  
 // 1.权值越高，查询结果越靠前。  
 // 2.lucene4.0以后不能对doc加权  
 // 3.只能对TextField加权  
 if** (**emails**[**i**].indexOf(**"qq.com"**) **!= -1**) **{  
 field**.setBoost(**2.0f**);  
 **} else if** (**emails**[**i**].indexOf(**"sina.com"**) **!= -1**) **{  
 field**.setBoost(**1.5f**);  
 **} else if** (**emails**[**i**].indexOf(**"google"**) **!= -1**) **{  
 field**.setBoost(**3.5f**);  
 **}  
  
 // 对于内容只索引不存储  
 doc**.add(**new** TextField(**"content"**, **contents**[**i**], **Field**.Store.***NO***));  
 **iw**.addDocument(**doc**);  
 **}  
 iw**.close();  
 **} catch** (**IOException e**) **{  
 e**.printStackTrace();  
 **}  
 }  
  
  
 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 查询(Read) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
//查询所有索引数目 maxDoc()  
 @Test  
 public void** readIndex() **{  
 try {** IndexReader **ir =** DirectoryReader.open(**directory**);  
 **System**.***out***.println(**"max num:" + ir**.**maxDoc**());  
 **System**.***out***.println(**"index num:" + ir**.**numDocs**());  
 **// 删除了的索引数 4.X版本后取消了恢复删除  
 System**.***out***.println(**"delete index num:" + ir**.numDeletedDocs());  
 **ir**.close();  
 **} catch** (**IOException e**) **{  
 e**.printStackTrace();  
 **}  
 }  
  
  
 // 根据条件查找索引  
 @Test  
 public void** queryIndex() **{  
 try {** IndexReader **ir =** DirectoryReader.open(**directory**);  
 **// 搜索器  
 IndexSearcher searcher = new** IndexSearcher(**ir**);  
 **// 查询哪个字段  
 QueryParser parse = new** QueryParser(**"email"**, **analyzer**);  
 **// 查询关键字** Query **query = parse**.parse(**"zs"**);  
 **TopDocs topDocs = searcher**.search(**query**, **1000**);  
  
 **// 碰撞结果  
 ScoreDoc**[] **hits = topDocs**.**scoreDocs**;  
  
 **for** (**int i = 0**; **i < hits**.**length**; **i++**) **{  
 ScoreDoc hit = hits**[**i**];  
 **Document hitDoc = searcher**.doc(**hit**.**doc**);  
 **// 结果按照得分来排序。主要由 关键字的个数和权值来决定  
 System**.***out***.println(**"(" + hit**.**doc + "-" + hit**.**score + ")" +  
 "id:" + hitDoc**.get(**"id"**) **+  
 " name:" + hitDoc**.get(**"name"**) **+  
 " email:" + hitDoc**.get(**"email"**) **+  
 " content:" + hitDoc**.get(**"content"**));  
 **}  
 ir**.close();  
 **} catch** (**IOException e**) **{  
 e**.printStackTrace();  
 **} catch** (**ParseException e**) **{  
 e**.printStackTrace();  
 **}  
 }  
  
  
 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 索引更新(Update) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
//原理是删除查询到的document 再增加一个新创建的document  
 @Test  
 public void** updateIndex() **{  
 IndexWriterConfig conf = new** IndexWriterConfig(**analyzer**);  
 **try {  
 IndexWriter iw = new** IndexWriter(**directory**, **conf**);  
 **Term term = new** Term(**"id"**, **"3"**);  
 **Document doc = new** Document();  
 **doc**.add(**new** StringField(**"id"**, **"9"**, **Field**.Store.***YES***));  
 **doc**.add(**new** StringField(**"name"**, **"lsup"**, **Field**.Store.***YES***));  
 **doc**.add(**new** StringField(**"email"**, **"liuzongyang@qq.com"**, **Field**.Store.***YES***));  
  
 **//加权 索引排序结果按照得分来排序。主要由关键字的个数和权值来决定  
 Field boostField = new** TextField(**"content"**, **contents**[**1**], **Field**.Store.***YES***);  
 **doc**.add(**boostField**);  
  
 **boostField**.setBoost(**5f**);  
 **// 更新的时候，会把原来那个索引删掉，重新生成一个索引  
 iw**.updateDocument(**term**, **doc**);  
  
 **iw**.commit();  
 **iw**.close();  
  
 **} catch** (**IOException e**) **{  
 e**.printStackTrace();  
 **}  
 }  
  
  
 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 索引删除(Delete) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
 @Test  
 public void** deleteIndex() **{  
 IndexWriterConfig conf = new** IndexWriterConfig(**analyzer**);  
 **try {  
 IndexWriter iw = new** IndexWriter(**directory**, **conf**);  
 **// Term[] terms = new Term[2];  
 // Term term = new Term("id", "3");  
 // terms[0] = term;  
 // term = new Term("id", "3");  
 // terms[1] = term;  
 // 将id为 1和3的索引删除。  
 // iw.deleteDocuments(term);  
 // 也可以传一个Query数组对象，将Query查找的结果删除。  
 QueryParser parse = new** QueryParser(**"id"**, **analyzer**);  
 **// 查询关键字** Query **query = parse**.parse(**"1"**);  
 **iw**.deleteDocuments(**query**);  
  
 **// deleteDocuments  
 iw**.commit();  
 **iw**.close();  
 **} catch** (**Exception e**) **{  
 e**.printStackTrace();  
 **}  
 }  
}**