**lucene的各种Field及其排序**

原创 2015年05月06日 18:27:57

**Lucene的Field说明**

Lucene存储对象是以document为存储单元，对象中相关的属性值则存放到Field中；

lucene中所有Field都是IndexableField接口的实现

**[html]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. org.apache.lucene.index.IndexableField

4. Represents a single field for indexing. IndexWriter consumes Iterable<IndexableField> as a document.

IndexableField接口提供了一些方法，主要是对field相关属性的获取，包括

**[html]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. /\*\* 获取field的名称 \*/
2. public String name();

**[java]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. /\*\* 获取field的类型fieldType \*/
2. public IndexableFieldType fieldType();

**[java]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. /\*\*
2. \*获取当前field的权重（评分值）
3. \*只有Field有评分的概念，如果我们想对document进行评分值的设定
4. \*必须预先对document中对应的field值进行评分设设定
5. \*/
6. public float boost();

**[java]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. /\*\* 如果此Filed为二进制类型的，返回相应的值\*/
2. public BytesRef binaryValue();

...

**[java]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. /\*\*
2. \* 创建一个用户索引此Field的TokenStream
3. \*/
4. public TokenStream tokenStream(Analyzer analyzer, TokenStream reuse) throws IOException;

所有的Field均是org.apache.lucene.document.Field的子类；

项目中我们常用的Field类型主要有IntField, LongField, FloatField, DoubleField, BinaryDocValuesField, NumericDocValuesField, SortedDocValuesField, StringField, TextField, StoredField.

|  |  |
| --- | --- |
| **lucene常见Field** | |
| **IntField** | **主要对int类型的字段进行存储，需要注意的是如果需要对InfField进行排序使用SortField.Type.INT来比较，如果进范围查询或过滤，需要采用NumericRangeQuery.newIntRange()** |
| LongField | 主要处理Long类型的字段的存储，排序使用SortField.Type.Long,如果进行范围查询或过滤利用NumericRangeQuery.newLongRange()，LongField常用来进行时间戳的排序，保存System.currentTimeMillions() |
| FloatField | 对Float类型的字段进行存储，排序采用SortField.Type.Float,范围查询采用NumericRangeQuery.newFloatRange() |
| BinaryDocVluesField | 只存储不共享值，如果需要共享值可以用SortedDocValuesField |
| NumericDocValuesField | 用于数值类型的Field的排序(预排序)，需要在要排序的field后添加一个同名的NumericDocValuesField |
| SortedDocValuesField | 用于String类型的Field的排序，需要在StringField后添加同名的SortedDocValuesField |
| StringField | 用户String类型的字段的存储，StringField是只索引不分词 |
| TextField | 对String类型的字段进行存储，TextField和StringField的不同是TextField既索引又分词 |
| StoredField | 存储Field的值，可以用IndexSearcher.doc和IndexReader.document来获取此Field和存储的值 |

**IntField使用**

**[java]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. package com.lucene.field;
3. import java.io.IOException;

6. import org.apache.lucene.document.Document;
7. import org.apache.lucene.document.Field;
8. import org.apache.lucene.document.IntField;
9. import org.apache.lucene.document.NumericDocValuesField;
10. import org.apache.lucene.index.IndexWriter;
11. import org.apache.lucene.search.IndexSearcher;
12. import org.apache.lucene.search.MatchAllDocsQuery;
13. import org.apache.lucene.search.Query;
14. import org.apache.lucene.search.ScoreDoc;
15. import org.apache.lucene.search.Sort;
16. import org.apache.lucene.search.SortField;
17. import org.apache.lucene.search.TopFieldDocs;
18. import org.junit.Test;
20. import com.lucene.index.IndexUtil;
21. import com.lucene.search.SearchUtil;
23. public class IntFieldTest {
24. /\*\*
25. \* 保存一个intField
26. \*/
27. @Test
28. public void testIndexIntFieldStored() {
29. Document document = new Document();
30. document.add(new IntField("intValue", 30, Field.Store.YES));
31. //要排序必须加同名的field，且类型为NumericDocValuesField
32. document.add(new NumericDocValuesField("intValue", 30));
33. Document document1 = new Document();
34. document1.add(new IntField("intValue", 40, Field.Store.YES));
35. document1.add(new NumericDocValuesField("intValue", 40));
36. IndexWriter writer = null;
37. try {
38. writer = IndexUtil.getIndexWriter("intFieldPath", false);
39. writer.addDocument(document);
40. writer.addDocument(document1);
42. } catch (IOException e) {
43. // TODO Auto-generated catch block
44. e.printStackTrace();
45. }finally{
46. try {
47. writer.commit();
48. writer.close();
49. } catch (IOException e) {
50. // TODO Auto-generated catch block
51. e.printStackTrace();
52. }
53. }
54. }
55. /\*\*
56. \* 测试intField排序
57. \*/
58. @Test
59. public void testIntFieldSort(){
60. try {
61. IndexSearcher searcher = SearchUtil.getIndexSearcher("intFieldPath", null);
62. //构建排序字段
63. SortField[] sortField = new SortField[1];
64. sortField[0] = new SortField("intValue",SortField.Type.INT,true);
65. Sort sort = new Sort(sortField);
66. //查询所有结果
67. Query query = new MatchAllDocsQuery();
68. TopFieldDocs docs = searcher.search(query, 2, sort);
69. ScoreDoc[] scores = docs.scoreDocs;
70. //遍历结果
71. for (ScoreDoc scoreDoc : scores) {
72. System.out.println(searcher.doc(scoreDoc.doc));;
73. }
74. //searcher.search(query, results);
75. } catch (IOException e) {
76. // TODO Auto-generated catch block
77. e.printStackTrace();
78. }
79. }

82. }

测试排序结果如下

**[html]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. Document<stored<intValue:40>>
2. Document<stored<intValue:30>>

如果修改NumericDocValuesField对应的值，结果会随着其值的大小而改变

**LongField使用**

**[java]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. package com.lucene.field;
3. import java.io.IOException;
5. import org.apache.lucene.document.Document;
6. import org.apache.lucene.document.Field;
7. import org.apache.lucene.document.LongField;
8. import org.apache.lucene.document.NumericDocValuesField;
9. import org.apache.lucene.index.IndexWriter;
10. import org.apache.lucene.search.IndexSearcher;
11. import org.apache.lucene.search.MatchAllDocsQuery;
12. import org.apache.lucene.search.Query;
13. import org.apache.lucene.search.ScoreDoc;
14. import org.apache.lucene.search.Sort;
15. import org.apache.lucene.search.SortField;
16. import org.apache.lucene.search.TopFieldDocs;
17. import org.junit.Test;
19. import com.lucene.index.IndexUtil;
20. import com.lucene.search.SearchUtil;
22. public class LongFieldTest {
24. /\*\*
25. \* 保存一个longField
26. \*/
27. @Test
28. public void testIndexLongFieldStored() {
29. Document document = new Document();
30. document.add(new LongField("longValue", 50L, Field.Store.YES));
31. document.add(new NumericDocValuesField("longValue", 50L));
32. Document document1 = new Document();
33. document1.add(new LongField("longValue", 80L, Field.Store.YES));
34. document1.add(new NumericDocValuesField("longValue", 80L));
35. IndexWriter writer = null;
36. try {
37. writer = IndexUtil.getIndexWriter("longFieldPath", false);
38. writer.addDocument(document);
39. writer.addDocument(document1);
41. } catch (IOException e) {
42. // TODO Auto-generated catch block
43. e.printStackTrace();
44. }finally{
45. try {
46. writer.commit();
47. writer.close();
48. } catch (IOException e) {
49. // TODO Auto-generated catch block
50. e.printStackTrace();
51. }
52. }
53. }
54. /\*\*
55. \* 测试longField排序
56. \*/
57. @Test
58. public void testLongFieldSort(){
59. try {
60. IndexSearcher searcher = SearchUtil.getIndexSearcher("longFieldPath", null);
61. //构建排序字段
62. SortField[] sortField = new SortField[1];
63. sortField[0] = new SortField("longValue",SortField.Type.LONG,true);
64. Sort sort = new Sort(sortField);
65. //查询所有结果
66. Query query = new MatchAllDocsQuery();
67. TopFieldDocs docs = searcher.search(query, 2, sort);
68. ScoreDoc[] scores = docs.scoreDocs;
69. //遍历结果
70. for (ScoreDoc scoreDoc : scores) {
71. //System.out.println(searcher.doc(scoreDoc.doc));;
72. Document doc = searcher.doc(scoreDoc.doc);
73. System.out.println(doc.getField("longValue").numericValue());
74. }
75. //searcher.search(query, results);
76. } catch (IOException e) {
77. // TODO Auto-generated catch block
78. e.printStackTrace();
79. }
80. }
81. }

运行结果如下：

**[html]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. Document<stored<longValue:80>>
2. Document<stored<longValue:50>>

**FloatField使用**

**[java]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. package com.lucene.field;
3. import java.io.IOException;
5. import org.apache.lucene.document.Document;
6. import org.apache.lucene.document.Field;
7. import org.apache.lucene.document.FloatDocValuesField;
8. import org.apache.lucene.document.FloatField;
9. import org.apache.lucene.index.IndexWriter;
10. import org.apache.lucene.search.IndexSearcher;
11. import org.apache.lucene.search.MatchAllDocsQuery;
12. import org.apache.lucene.search.Query;
13. import org.apache.lucene.search.ScoreDoc;
14. import org.apache.lucene.search.Sort;
15. import org.apache.lucene.search.SortField;
16. import org.apache.lucene.search.TopFieldDocs;
17. import org.junit.Test;
19. import com.lucene.index.IndexUtil;
20. import com.lucene.search.SearchUtil;
22. public class FloatFieldTest {
24. /\*\*
25. \* 保存一个floatField
26. \*/
27. @Test
28. public void testIndexFloatFieldStored() {
29. Document document = new Document();
30. document.add(new FloatField("floatValue", 9.1f, Field.Store.YES));
31. document.add(new FloatDocValuesField("floatValue", 82.0f));
32. Document document1 = new Document();
33. document1.add(new FloatField("floatValue", 80.1f, Field.Store.YES));
34. document1.add(new FloatDocValuesField("floatValue", 80.1f));
35. IndexWriter writer = null;
36. try {
37. writer = IndexUtil.getIndexWriter("floatFieldPath", false);
38. writer.addDocument(document);
39. writer.addDocument(document1);
41. } catch (IOException e) {
42. // TODO Auto-generated catch block
43. e.printStackTrace();
44. }finally{
45. try {
46. writer.commit();
47. writer.close();
48. } catch (IOException e) {
49. // TODO Auto-generated catch block
50. e.printStackTrace();
51. }
52. }
53. }
54. /\*\*
55. \* 测试intField排序
56. \*/
57. @Test
58. public void testFloatFieldSort(){
59. try {
60. IndexSearcher searcher = SearchUtil.getIndexSearcher("floatFieldPath", null);
61. //构建排序字段
62. SortField[] sortField = new SortField[1];
63. sortField[0] = new SortField("floatValue",SortField.Type.FLOAT,true);
64. Sort sort = new Sort(sortField);
65. //查询所有结果
66. Query query = new MatchAllDocsQuery();
67. TopFieldDocs docs = searcher.search(query, 2, sort);
68. ScoreDoc[] scores = docs.scoreDocs;
69. //遍历结果
70. for (ScoreDoc scoreDoc : scores) {
71. //System.out.println(searcher.doc(scoreDoc.doc));;
72. Document doc = searcher.doc(scoreDoc.doc);
73. System.out.println(doc.getField("floatValue").numericValue());
74. }
75. //searcher.search(query, results);
76. } catch (IOException e) {
77. // TODO Auto-generated catch block
78. e.printStackTrace();
79. }
80. }
81. }

结果如下：

**[html]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. Document<stored<floatValue:9.1>>
2. Document<stored<floatValue:80.1>>

**BinaryDocValuesField使用**

**[java]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. package com.lucene.field;
3. import java.io.IOException;
5. import org.apache.lucene.document.BinaryDocValuesField;
6. import org.apache.lucene.document.Document;
7. import org.apache.lucene.document.Field;
8. import org.apache.lucene.document.FloatDocValuesField;
9. import org.apache.lucene.document.FloatField;
10. import org.apache.lucene.document.IntField;
11. import org.apache.lucene.document.LongField;
12. import org.apache.lucene.document.NumericDocValuesField;
13. import org.apache.lucene.index.IndexWriter;
14. import org.apache.lucene.search.IndexSearcher;
15. import org.apache.lucene.search.MatchAllDocsQuery;
16. import org.apache.lucene.search.Query;
17. import org.apache.lucene.search.ScoreDoc;
18. import org.apache.lucene.search.Sort;
19. import org.apache.lucene.search.SortField;
20. import org.apache.lucene.search.TopFieldDocs;
21. import org.apache.lucene.util.BytesRef;
22. import org.junit.Test;
24. import com.lucene.index.IndexUtil;
25. import com.lucene.search.SearchUtil;
27. public class BinaryDocValuesFieldTest {
29. /\*\*
30. \* 保存一个BinaryDocValuesField
31. \*/
32. @Test
33. public void testIndexLongFieldStored() {
34. Document document = new Document();
35. document.add(new BinaryDocValuesField("binaryValue",new BytesRef("1234".getBytes())));
36. Document document1 = new Document();
37. document1.add(new BinaryDocValuesField("binaryValue",new BytesRef("2345".getBytes())));
38. IndexWriter writer = null;
39. try {
40. writer = IndexUtil.getIndexWriter("binaryValueFieldPath", false);
41. writer.addDocument(document);
42. writer.addDocument(document1);
44. } catch (IOException e) {
45. // TODO Auto-generated catch block
46. e.printStackTrace();
47. }finally{
48. try {
49. writer.commit();
50. writer.close();
51. } catch (IOException e) {
52. // TODO Auto-generated catch block
53. e.printStackTrace();
54. }
55. }
56. }
57. /\*\*
58. \* 测试BinaryDocValuesField排序
59. \*/
60. @Test
61. public void testBinaryDocValuesFieldSort(){
62. try {
63. IndexSearcher searcher = SearchUtil.getIndexSearcher("binaryValueFieldPath", null);
64. //构建排序字段
65. SortField[] sortField = new SortField[1];
66. sortField[0] = new SortField("binaryValue",SortField.Type.STRING\_VAL,true);
67. Sort sort = new Sort(sortField);
68. //查询所有结果
69. Query query = new MatchAllDocsQuery();
70. TopFieldDocs docs = searcher.search(query, 2, sort);
71. ScoreDoc[] scores = docs.scoreDocs;
72. //遍历结果
73. for (ScoreDoc scoreDoc : scores) {
74. //System.out.println(searcher.doc(scoreDoc.doc));;
75. Document doc = searcher.doc(scoreDoc.doc);
76. System.out.println(doc);
77. //System.out.println(doc.getField("binaryValue").numericValue());
78. }
79. //searcher.search(query, results);
80. } catch (IOException e) {
81. // TODO Auto-generated catch block
82. e.printStackTrace();
83. }
84. }
85. }

运行结果：

**[html]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. Document<>
2. Document<>

为什么这样呢，这是跟BinaryDocValuesField的特性决定的，只索引不存值！

**StringField使用**

**[java]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. package com.lucene.field;
3. import java.io.IOException;
5. import org.apache.lucene.document.BinaryDocValuesField;
6. import org.apache.lucene.document.Document;
7. import org.apache.lucene.document.Field;
8. import org.apache.lucene.document.FloatDocValuesField;
9. import org.apache.lucene.document.FloatField;
10. import org.apache.lucene.document.IntField;
11. import org.apache.lucene.document.LongField;
12. import org.apache.lucene.document.NumericDocValuesField;
13. import org.apache.lucene.document.SortedDocValuesField;
14. import org.apache.lucene.document.StringField;
15. import org.apache.lucene.index.IndexWriter;
16. import org.apache.lucene.search.IndexSearcher;
17. import org.apache.lucene.search.MatchAllDocsQuery;
18. import org.apache.lucene.search.Query;
19. import org.apache.lucene.search.ScoreDoc;
20. import org.apache.lucene.search.Sort;
21. import org.apache.lucene.search.SortField;
22. import org.apache.lucene.search.TopFieldDocs;
23. import org.apache.lucene.util.BytesRef;
24. import org.junit.Test;
26. import com.lucene.index.IndexUtil;
27. import com.lucene.search.SearchUtil;
29. public class StringFieldTest {
31. /\*\*
32. \* 保存一个StringField
33. \*/
34. @Test
35. public void testIndexLongFieldStored() {
36. Document document = new Document();
37. document.add(new StringField("stringValue","12445", Field.Store.YES));
38. document.add(new SortedDocValuesField("stringValue", new BytesRef("12445".getBytes())));
39. Document document1 = new Document();
40. document1.add(new StringField("stringValue","23456", Field.Store.YES));
41. document1.add(new SortedDocValuesField("stringValue", new BytesRef("23456".getBytes())));
42. IndexWriter writer = null;
43. try {
44. writer = IndexUtil.getIndexWriter("stringFieldPath", false);
45. writer.addDocument(document);
46. writer.addDocument(document1);
48. } catch (IOException e) {
49. // TODO Auto-generated catch block
50. e.printStackTrace();
51. }finally{
52. try {
53. writer.commit();
54. writer.close();
55. } catch (IOException e) {
56. // TODO Auto-generated catch block
57. e.printStackTrace();
58. }
59. }
60. }
61. /\*\*
62. \* 测试StringField排序
63. \*/
64. @Test
65. public void testStringFieldSort(){
66. try {
67. IndexSearcher searcher = SearchUtil.getIndexSearcher("stringFieldPath", null);
68. //构建排序字段
69. SortField[] sortField = new SortField[1];
70. sortField[0] = new SortField("stringVal",SortField.Type.STRING,true);
71. Sort sort = new Sort(sortField);
72. //查询所有结果
73. Query query = new MatchAllDocsQuery();
74. TopFieldDocs docs = searcher.search(query, 2, sort);
75. ScoreDoc[] scores = docs.scoreDocs;
76. //遍历结果
77. for (ScoreDoc scoreDoc : scores) {
78. //System.out.println(searcher.doc(scoreDoc.doc));;
79. Document doc = searcher.doc(scoreDoc.doc);
80. System.out.println(doc);
81. //System.out.println(doc.getField("binaryValue").numericValue());
82. }
83. //searcher.search(query, results);
84. } catch (IOException e) {
85. // TODO Auto-generated catch block
86. e.printStackTrace();
87. }
88. }
89. }

运行结果如下：

**[html]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. Document<stored,indexed,tokenized,omitNorms,indexOptions=DOCS<stringValue:12445>>
2. Document<stored,indexed,tokenized,omitNorms,indexOptions=DOCS<stringValue:23456>>

**TextField使用**

**[java]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. package com.lucene.field;
3. import java.io.IOException;
5. import org.apache.lucene.document.BinaryDocValuesField;
6. import org.apache.lucene.document.Document;
7. import org.apache.lucene.document.Field;
8. import org.apache.lucene.document.FloatDocValuesField;
9. import org.apache.lucene.document.FloatField;
10. import org.apache.lucene.document.IntField;
11. import org.apache.lucene.document.LongField;
12. import org.apache.lucene.document.NumericDocValuesField;
13. import org.apache.lucene.document.SortedDocValuesField;
14. import org.apache.lucene.document.StringField;
15. import org.apache.lucene.document.TextField;
16. import org.apache.lucene.index.IndexWriter;
17. import org.apache.lucene.search.IndexSearcher;
18. import org.apache.lucene.search.MatchAllDocsQuery;
19. import org.apache.lucene.search.Query;
20. import org.apache.lucene.search.ScoreDoc;
21. import org.apache.lucene.search.Sort;
22. import org.apache.lucene.search.SortField;
23. import org.apache.lucene.search.TopFieldDocs;
24. import org.apache.lucene.util.BytesRef;
25. import org.junit.Test;
27. import com.lucene.index.IndexUtil;
28. import com.lucene.search.SearchUtil;
30. public class TextFieldTest {
32. /\*\*
33. \* 保存一个StringField
34. \*/
35. @Test
36. public void testIndexLongFieldStored() {
37. Document document = new Document();
38. document.add(new TextField("textValue","12345", Field.Store.YES));
39. document.add(new SortedDocValuesField("textValue", new BytesRef("12345".getBytes())));
40. Document document1 = new Document();
41. document1.add(new TextField("textValue","23456", Field.Store.YES));
42. document1.add(new SortedDocValuesField("textValue", new BytesRef("23456".getBytes())));
43. IndexWriter writer = null;
44. try {
45. writer = IndexUtil.getIndexWriter("textFieldPath", false);
46. writer.addDocument(document);
47. writer.addDocument(document1);
49. } catch (IOException e) {
50. // TODO Auto-generated catch block
51. e.printStackTrace();
52. }finally{
53. try {
54. writer.commit();
55. writer.close();
56. } catch (IOException e) {
57. // TODO Auto-generated catch block
58. e.printStackTrace();
59. }
60. }
61. }
62. /\*\*
63. \* 测试StringField排序
64. \*/
65. @Test
66. public void testStringFieldSort(){
67. try {
68. IndexSearcher searcher = SearchUtil.getIndexSearcher("textFieldPath", null);
69. //构建排序字段
70. SortField[] sortField = new SortField[1];
71. sortField[0] = new SortField("textValue",SortField.Type.STRING,true);
72. Sort sort = new Sort(sortField);
73. //查询所有结果
74. Query query = new MatchAllDocsQuery();
75. TopFieldDocs docs = searcher.search(query, 2, sort);
76. ScoreDoc[] scores = docs.scoreDocs;
77. //遍历结果
78. for (ScoreDoc scoreDoc : scores) {
79. //System.out.println(searcher.doc(scoreDoc.doc));;
80. Document doc = searcher.doc(scoreDoc.doc);
81. System.out.println(doc);
82. //System.out.println(doc.getField("binaryValue").numericValue());
83. }
84. //searcher.search(query, results);
85. } catch (IOException e) {
86. // TODO Auto-generated catch block
87. e.printStackTrace();
88. }
89. }
90. }

运行结果如下：

**[html]** [view plain](http://blog.csdn.net/wuyinggui10000/article/details/45538155) [copy](http://blog.csdn.net/wuyinggui10000/article/details/45538155)

1. Document<stored,indexed,tokenized<textValue:23456>>
2. Document<stored,indexed,tokenized<textValue:12345>>

**源码下载地址**

[lucene field使用源码](http://download.csdn.net/detail/wuyinggui10000/8669987)