

## Introduction

This class acts as a core component which reads/searches indexes during searching process.

## Class declaration

Following is the declaration for **org.apache.lucene.search.IndexSearcher** class:

```
public class IndexSearcher
    extends Searcher
```

## Field

Following are the fields for **org.apache.lucene.index.IndexWriter** class:

- **protected int[] docStarts**
- **protected IndexReader[] subReaders**
- **protected IndexSearcher[] subSearchers**

## Class constructors

### S.N. Constructor & Description

- |   |  |
|---|--|
| 1 | <b>IndexSearcherDirectorypath</b><br>Deprecated. Use <code>IndexSearcherIndexReader</code> instead.  |
| 2 | <b>IndexSearcherDirectorypath, booleanreadOnly</b><br>Deprecated. Use <code>IndexSearcherIndexReader</code> instead.   |
| 3 | <b>IndexSearcherIndexReaderr</b><br>Creates a searcher searching the provided index.   |
| 4 | <b>IndexSearcherIndexReaderr, ExecutorServiceexecutor</b><br>Runs searches for each segment separately, using the provided <code>ExecutorService</code> .  |
| 5 | <b>IndexSearcherIndexReaderreader, IndexReader[]subReaders, int[]docStarts</b><br>Expert: directly specify the reader, subReaders and their docID starts.  |
| 6 | <b>IndexSearcherIndexReaderreader, IndexReader[]subReaders, int[]docStarts, ExecutorServiceexecutor</b><br>Expert: directly specify the reader, subReaders and their docID starts, and an <code>ExecutorService</code> . |

## Class methods

### S.N. Method & Description

- 1 **void close**  
Note that the underlying IndexReader is not closed, if IndexSearcher was constructed with IndexSearcher*IndexReader*.
- 2 **Weight createNormalizedWeightQueryquery**  
Creates a normalized weight for a top-level Query.
- 3 **Document docintdocID**  
Returns the stored fields of document i.
- 4 **Document docintdocID, FieldSelectorfieldSelector**  
Get the Document at the nth position.
- 5 **int docFreqTermterm**  
Returns total docFreq for this term.
- 6 **Explanation explainQueryquery, intdoc**  
Returns an Explanation that describes how doc scored against query.
- 7 **Explanation explainWeightweight, intdoc**  
Expert: low-level implementation method Returns an Explanation that describes how doc scored against weight.
- 8 **protected void gatherSubReaders(List allSubReaders, IndexReader r)**
- 9 **IndexReader getIndexReader**  
Return the IndexReader this searches.
- 10 **Similarity getSimilarity**  
Expert: Return the Similarity implementation used by this Searcher.
- 11 **IndexReader[] getSubReaders**

Returns the atomic subReaders used by this searcher.

12

**int maxDoc**

Expert: Returns one greater than the largest possible document number.

13

**Query rewrite***Queryoriginal*

Expert: called to re-write queries into primitive queries.

14

**void search***Queryquery, Collectorresults*

Lower-level search API.

15

**void search***Queryquery, Filterfilter, Collectorresults*

Lower-level search API.

16

**TopDocs search***Queryquery, Filterfilter, intn*

Finds the top n hits for query, applying filter if non-null.

17

**TopFieldDocs search***Queryquery, Filterfilter, intn, Sortsort*

Search implementation with arbitrary sorting.

18

**TopDocs search***Queryquery, intn*

Finds the top n hits for query.

19

**TopFieldDocs search***Queryquery, intn, Sortsort*

Search implementation with arbitrary sorting and no filter.

20

**void search***Weightweight, Filterfilter, Collectorcollector*

Lower-level search API.

21

**TopDocs search***Weightweight, Filterfilter, intnDocs*

Expert: Low-level search implementation.

22

**TopFieldDocs search***Weightweight, Filterfilter, intnDocs, Sortsort*

Expert: Low-level search implementation with arbitrary sorting.

23

**protected TopFieldDocs search***Weightweight, Filterfilter, intnDocs, Sortsort, booleanfillFields*

Just like `searchWeight, Filter, int, Sort`, but you choose whether or not the fields in the returned `FieldDoc` instances should be set by specifying `fillFields`.

- 24 **protected TopDocs searchWeightweight, Filterfilter, ScoreDocafter, intnDocs**  
Expert: Low-level search implementation.
- 25 **TopDocs searchAfterScoreDocafter, Queryquery, Filterfilter, intn**  
Finds the top n hits for query, applying filter if non-null, where all results are after a previous result *after*.
- 26 **TopDocs searchAfterScoreDocafter, Queryquery, intn**  
Finds the top n hits for query where all results are after a previous result *after*.
- 27 **void setDefaultFieldSortScoringbooleandoTrackScores, booleandoMaxScore**  
By default, no scores are computed when sorting by field *using* `search(Query, Filter, int, Sort)`.
- 28 **void setSimilaritySimilaritysimilarity**  
Expert: Set the Similarity implementation used by this Searcher.
- 29 **String toString**

## Methods inherited

This class inherits methods from the following classes:

- `org.apache.lucene.search.Searcher`
- `java.lang.Object`

Processing math: 100%