

Apache Lucene - Basic Demo Sources Walkthrough

Andrew C. Oliver

Table of contents

1 About the Code.....	2
2 Location of the source (developers/deployers).....	2
3 index.jsp (developers/deployers).....	2
4 header.jsp (developers/deployers).....	2
5 results.jsp (developers).....	2
6 More sources (developers).....	3
7 Where to go from here? (everyone!).....	3
8 When to contact the Author.....	4

1. About the Code

In this section we walk through the sources behind the basic Lucene Web Application demo: where to find them, their parts and their function. This section is intended for Java developers wishing to understand how to use Lucene in their applications or for those involved in deploying web applications based on Lucene.

2. Location of the source (developers/deployers)

Relative to the directory created when you extracted Lucene or retrieved it from Subversion, you should see a directory called `src` which in turn contains a directory called `jsp`. This is the root for all of the Lucene web demo.

Within this directory you should see `index.jsp`. Bring this up in `vi` or your editor of choice.

3. `index.jsp` (developers/deployers)

This `jsp` page is pretty boring by itself. All it does is include a header, display a form and include a footer. If you look at the form, it has two fields: `query` (where you enter your search criteria) and `maxresults` where you specify the number of results per page. By the structure of this JSP it should be easy to customize it without even editing this particular file. You could simply change the header and footer. Let's look at the `header.jsp` (located in the same directory) next.

4. `header.jsp` (developers/deployers)

The header is also very simple by itself. The only thing it does is include the `configuration.jsp` (which you looked at in the last section of this guide) and set the title and a brief header. This would be a good place to put your own custom HTML to "pretty" things up a bit. We won't cover the footer because all it does is display the footer and close your tags. Let's look at the `results.jsp`, the meat of this application, next.

5. `results.jsp` (developers)

Most of the functionality lies in `results.jsp`. Much of it is for paging the search results, which we'll not cover here as it's commented well enough. The first thing in this page is the actual imports for the Lucene classes and Lucene demo classes. These classes are loaded from the jars included in the `WEB-INF/lib` directory in the `luceneweb.war` file.

You'll notice that this file includes the same header and footer as `index.jsp`. From there it

constructs an `IndexSearcher` with the `indexLocation` that was specified in `configuration.jsp`. If there is an error of any kind in opening the index, it is displayed to the user and the boolean flag `error` is set to tell the rest of the sections of the jsp not to continue.

From there, this jsp attempts to get the search criteria, the start index (used for paging) and the maximum number of results per page. If the maximum results per page is not set or not valid then it and the start index are set to default values. If only the start index is invalid it is set to a default value. If the criteria isn't provided then a servlet error is thrown (it is assumed that this is the result of url tampering or some form of browser malfunction).

The jsp moves on to construct a `StandardAnalyzer` to analyze the search text. This matches the analyzer used during indexing (`IndexHTML`), which is generally recommended. This is passed to the `QueryParser` along with the criteria to construct a `Query` object. You'll also notice the string literal `"contents"` included. This specifies that the search should cover the `contents` field and not the `title`, `url` or some other field in the indexed documents. If there is any error in constructing a `Query` object an error is displayed to the user.

In the next section of the jsp the `IndexSearcher` is asked to search given the query object. The results are returned in a collection called `hits`. If the length property of the `hits` collection is 0 (meaning there were no results) then an error is displayed to the user and the error flag is set.

Finally the jsp iterates through the `hits` collection, taking the current page into account, and displays properties of the `Document` objects we talked about in the first walkthrough. These objects contain "known" fields specific to their indexer (in this case `IndexHTML` constructs a document with `"url"`, `"title"` and `"contents"`).

Please note that in a real deployment of Lucene, it's best to instantiate `IndexSearcher` and `QueryParser` once, and then share them across search requests, instead of re-instantiating per search request.

6. More sources (developers)

There are additional sources used by the web app that were not specifically covered by either walkthrough. For example the HTML parser, the `IndexHTML` class and `HTMLDocument` class. These are very similar to the classes covered in the first example, with properties specific to parsing and indexing HTML. This is beyond our scope; however, by now you should feel like you're "getting started" with Lucene.

7. Where to go from here? (everyone!)

There are a number of things this demo doesn't do or doesn't do quite right. For instance, you may have noticed that documents in the root context are unreachable (unless you reconfigure Tomcat to support that context or redirect to it), anywhere where the directory doesn't quite match the context mapping, you'll have a broken link in your results. If you want to index non-local files or have some other needs this isn't supported, plus there may be security issues with running the indexing application from your webapps directory. There are a number of things left for you the developer to do.

In time some of these things may be added to Lucene as features (if you've got a good idea we'd love to hear it!), but for now: this is where you begin and the search engine/indexer ends. Lastly, one would assume you'd want to follow the above advice and customize the application to look a little more fancy than black on white with "Lucene Template" at the top. We'll see you on the Lucene Users' or Developers' mailing lists!

8. When to contact the Author

Please resist the urge to contact the authors of this document (without bribes of fame and fortune attached). First contact the mailing lists, taking care to Ask Questions The Smart Way. Certainly you'll get the most help that way as well. That being said, feedback, and modifications to this document and samples are ever so greatly appreciated. They are just best sent to the lists or posted as patches, so that everyone can share in them. Thanks for understanding!