FAQ

This is the Official Solr FAQ. There is also a SolrTerminology document that may be useful for understanding what some documentation means; as well as a Relevancy FAQ for addressing questions specific to how Relevancy Scoring works in Solr.

- General
  - What is Solr?
  - Are there Mailing lists for Solr?
  - How do you pronounce Solr?
  - What does Solr stand for?
  - Where did Solr come from?
  - Is Solr Stable? Is it "Production Quality?"
  - Is Solr vulnerable to any security exploits?
  - Is Solr Schema-less?
  - Is Solr just a wrapper around Lucene?
- Using
  - Do my applications have to be written in Java to use Solr?
  - What are the Requirements for running a Solr server?
  - How can I get started playing with Solr?
  - Solr Comes with Jetty, is Jetty the recommended Servlet Container to use when running Solr?
  - How do I change the logging levels/files/format ?
  - I POSTed some documents, why don't they show up when I search?
  - How can I delete all documents from my index?
  - How can I rebuild my index from scratch if I change my schema?
  - How do I reindex my data?
  - How can I update a specific field of an existing document?
  - How do I use copyField with wildcards?
  - Why does the request time out sometimes when doing commits?
  - Why don't International Characters Work?
  - Solr started, and I can POST documents to it, but the admin screen doesn't work
  - The Solr admin pages suddenly stop working and give a 404 error
  - What does "CorruptIndexException: Unknown format version" mean ?
  - What does "exceeded limit of maxWarmingSearchers=X" mean?
  - Why doesn't my index directory get smaller (immediately) when i delete documents? force a merge? optimize?
- Searching
  - Why Aren't Scores returned as a percentage? How Do I normalize Scores?
  - How to make the search use AND semantics by default rather than OR?
  - Why does 'foo AND -baz' match docs, but 'foo AND (-bar)' doesn't ?
  - How do I add full-text summaries to my search results?
  - I have set hl=true but no summaries are being output
  - I want to add basic category counts to my search results
  - How can I figure out why my documents are being ranked the way they are?
  - Why Isn't Sorting Working on my Text Fields?
  - My search returns too many / too little / unexpected results, how to debug?
  - How can I get ALL the matching documents back? ... How can I return an unlimited number of rows?
  - Can I use Lucene to access the index generated by SOLR?
  - Is there a limit on the number of keywords for a Solr query?
  - How can I efficently search for all documents that contain a value in fieldX ?
- Solr Cloud
  - Can I reuse the same ZooKeeper cluster with other applications, or multiple SolrCloud clusters?
  - Why doesn't SolrCloud automatically create replicas when I add nodes?
- Performance
  - How fast is indexing?
  - How can indexing be accelerated?
  - How can I speed up facet counts?
  - What does "PERFORMANCE WARNING: Overlapping onDeckSearchers=X" mean in my logs?
  - Why is the QTime Solr returns lower then the amount of time I'm measuring in my client?
- Developing
  - Where can I find the latest and Greatest Code?
  - Where can I get the javadocs for the classes?
  - How can I help?
  - How can I submit bug reports, bug fixes or new features?
  - How do I apply patches from JIRA issues?
  - I can't compile Solr, ant says "JUnit not found" or "Could not create task or type of type: junit"
  - How can I start the example application in Debug mode?
  - Tagging using SOLR
  - How can I get hold of HttpServletRequest object in custom first-component?

General

What is Solr?

Solr is a stand alone enterprise search server which applications communicate with using XML and HTTP to index documents, or execute searches. Solr supports a rich schema specification that allows for a wide range of flexibility in dealing with different document fields, and has an extensive search plugin API for developing custom search behavior.

For more information please read this overview of Solr features.

Are there Mailing lists for Solr?

Yes there are several Solr email lists.

Here are some guidelines for effectively using the email lists Getting the most out of the email lists.

How do you pronounce Solr?

It's pronounced the same as you would pronounce "Solar".

What does Solr stand for?

Solr is not an acronym.

Where did Solr come from?

"Solar" (with an A) was initially developed by CNET Networks as an in-house search platform beginning in late fall 2004. By summer 2005, CNET's product catalog was powered by Solar, and several other CNET applications soon followed. In January 2006 CNET Granted the existing code base to the ASF to become the "Solr" project. On January 17, 2007 Solr graduated from the Apache Incubator to become a Lucene subproject. In March 2010, The Solr and Lucene-java subprojects merged into a single project.

Is Solr Stable? Is it "Production Quality?"

Solr is currently being used to power search applications on several high traffic publicly accessible websites.

Is Solr vulnerable to any security exploits?

Every effort is made to ensure that Solr is not vulnerable to any known exploits. For specific information, see SolrVulnerabilities. Because Solr does not actually have any built-in security features, it should not be installed in a location that can be directly reached by anyone that cannot be trusted.

Is Solr Schema-less?

Yes. Solr does have a schema to define types, but it's a "free" schema in that you don't have to define all of your fields ahead of time. Using % 3CdynamicField /%3E declarations, you can configure field types based on field naming convention, and each document you index can have a different set of fields. Also, Solr supports a schemaless mode in which previously unseen fields' types are detected based on field values, and the resulting typed fields are automatically added to the schema.

Is Solr just a wrapper around Lucene?

No. Solr has been constantly innovating core search features since the beginning.

Using

Do my applications have to be written in Java to use Solr?

No.

Solr itself is a Java Application, but all interaction with Solr is done by POSTing messages over HTTP (in JSON, XML, CSV, or binary formats) to index documents and GETing search results back as JSON, XML, or a variety of other formats (Python, Ruby, PHP, CSV, binary, etc...)

What are the Requirements for running a Solr server?

Solr requires Java 1.5 and an Application server (such as Tomcat) which supports the Servlet 2.4 standard.

How can I get started playing with Solr?

There is an online tutorial as well as a demonstration configuration in SVN.

Solr Comes with Jetty, is Jetty the recommended Servlet Container to use when running Solr?

Prior to Solr 5.x, the Solr example app had Jetty in it just because at the time we set it up, Jetty was the simplest/smallest servlet container we found that could be run easily in a cross platform way (ie: "java -jar start.jar"). There was no implication implying that Solr runs better under Jetty, or that Jetty is only good enough for demos – it's just that Jetty made our demo setup easier. Since then, our test suite has grown to include items that actually start Jetty and run full integration tests, so we can be sure that Jetty works correctly. Other containers are not tested.

As of Solr 5.0, the .war file is a little bit harder to find and the startup scripts included with Solr in the bin directory are specifically designed to run Jetty. The documentation says that there is no longer any support for running Solr in a third-party container. This is technically not true - if you grab the war from server/webapps, the logging jars from server/lib/ext, and the log4j.properties file from server/resources, you can still deploy in a third-party container, but eventually this kind of deployment will no longer be possible. See WhyNoWar for a larger discussion about future plans where Solr will become a standalone application.

How do I change the logging levels/files/format ?

See SolrLogging

I POSTed some documents, why don't they show up when I search?

Documents that have been added to the index don't show up in search results until a commit is done (one way is to POST a <commit/> message to the XML update handler). e.g.

curl http://my.host/solr/update --data '<commit/>' -H 'Content-type:text/xml; charset=utf-8'

This allows you to POST many documents in succession and know that none of them will be visible to search clients until you have finished.

How can I delete all documents from my index?

Use the "match all docs" query in a delete by query command: `<delete><query>*:*</query></delete>`

You must also commit after running the delete so, to empty the index, run the following two commands:

```
curl http://localhost:8983/solr/update --data '<delete><query>*:*</query></delete>' -H 'Content-type:text/xml;
charset=utf-8'
curl http://localhost:8983/solr/update --data '<commit/>' -H 'Content-type:text/xml; charset=utf-8'
```

Another strategy would be to add two bookmarks in your browser:

```
http://localhost:8983/solr/update?stream.body=<delete><query>*:*</query></delete>
http://localhost:8983/solr/update?stream.body=<commit/>
```

And use those as you're developing to clear out the index as necessary.

How can I rebuild my index from scratch if I change my schema?

1. Use the "match all docs" query in a delete by query command before shutting down Solr: `<delete><query>*:*</query></delete>`
2. Stop your server
3. Change your schema.xml
4. Start your server
5. Re-Index your data

One can also delete all documents, change the schema.xml file, and then reload the core w/o shutting down Solr.

How do I reindex my data?

This deceptively simple question requires its own page: HowToReindex

How can I update a specific field of an existing document?

I want update a specific field in a document, is that possible? I only need to index one field for a specific document. Do I have to index all the document for this?

No, just the one document. Let's say you have a CMS and you edit one document. You will need to re-index this document only by using the the add solr statement for the whole document (not one field only).

In Lucene to update a document the operation is really a delete followed by an add. You will need to add the complete document as there is no such "update only a field" semantics in Lucene.

How do I use copyField with wildcards?

The `<copyField>` directive allows wildcards in the source, so that several fields can be copied into one destination field without having to specify them all individually. The dest field may by a full field name, or a wildcard expression. A common use case is something like:

```
<copyField source="*_t"  dest="text" />
```

This tells Solr to copy the contents of any field that ends in "_t" to the "text" field. This is particularly useful when you have a large, and possibly changing, set of fields you want to index into a single field. With the example above, you could start indexing fields like "description_t", "editorial_review_t", and so on, and all their content would be indexed in the "text" field. It's important in this example that the "text" field be defined in schema.xml as multiValued since you intend to copy multiple sources into the single destination.

Note that you can use the wildcard copyField syntax with or without similar dynamicField declarations. Thus you could choose to index the "description_t", "editorial_review_t" fields individually with a dynamicField like

```
<dynamicField name="*_t" type="text" indexed="true" stored="false" />
```

but you don't have to if you don't want to. You could even mix and match across different dynamic fields by doing something like

```
<dynamicField name="*_i_t" type="text" indexed="true" stored="false" />
<copyField source="*_t"  dest="text" />
```

Now, as you add fields, you can give them names ending in "_i_t" if you want them indexed seperately, and stored in the main "text" field, and "_t" without the "_i" if you just want them indexed in "text" but not individually.

Why does the request time out sometimes when doing commits?

Internally, Solr does nothing to time out any requests – it lets both updates and queries take however long they need to take to be processed fully. However, the servlet container being used to run Solr may impose arbitrary timeout limits on all requests. Please consult the documentation for your Serlvet container if you find that this value is too low.

(In Jetty, the relevant setting is "maxIdleTime" which is in milliseconds)

Why don't International Characters Work?

Solr can index any characters expressed in the UTF-8 charset (see SOLR-96). There are no known bugs with Solr's character handling, but there have been some reported issues with the way different application servers (and different versions of the same application server) treat incoming and outgoing multibyte characters. In particular, people have reported better success with Tomcat than with Jetty...

- "International Charsets in embedded XML" (Jetty 5.1)
- "Problem with surrogate characters in utf-8" (Jetty 6)

If you notice a problem with multibyte characters, the first step to ensure that it is not a true Solr bug would be to write a unit test that bypasses the application server directly using the AbstractSolrTestCase.

The most important points are:

- The document has to be indexed as UTF-8 encoded on the solr server. For example, if you send an ISO encoded document, then the special ISO characters get a byte added (screwing up the final encoding, only reindexing with UTF-8 can fix this).
- The client needs UTF-8 URL encoding when forwarding the search request to the solr server.
- The server needs to support UTF-8 query strings. See e.g. Solr with Apache Tomcat.

If you just forward doing:

```
String value = request.getParameter("q");
```

to get the query string, it can be that q got encoded in ISO and then solr will not return a search result.

One possible solution is:

```
String encoding = request.getCharacterEncoding();
if (null == encoding) {
  // Set your default encoding here
  request.setCharacterEncoding("UTF-8");
} else {
  request.setCharacterEncoding(encoding);
}
...
String value = request.getParameter("q");
```

Another possibility is to use java.net.URLDecoder/URLEncoder to transform all parameter value to UTF-8.

Solr started, and I can POST documents to it, but the admin screen doesn't work

The admin screens are implemented using JSPs which require a JDK (instead of just a JRE) to be compiled on the fly. If you encounter errors trying to load the admin pages, and the stack traces of these errors seem to relate to compilation of JSPs, make sure you have a JDK installed, and make sure it is the instance of java being used.

NOTE: Some Servlet Containers (like Tomcat5.5 and Jetty6) don't require a JDK for JSPs.

The Solr admin pages suddenly stop working and give a 404 error

See SOLR-118, this happens when using the default Jetty config provided by Solr, and having Jetty's work files in /tmp purged by the operating system.

Restarting Solr after creating a $(jetty.home)/work directory for Jetty's work files should solve the problem.

This might also be caused by starting two Solr instances on the same port and killing one, see Hoss's comment in SOLR-118.

What does "CorruptIndexException: Unknown format version" mean ?

This happens when the Lucene code in Solr used to read the index files from disk encounters index files in a format it doesn't recognize.

The most common cause is from using a version of Solr+Lucene that is older then the version used to create that index.

What does "exceeded limit of maxWarmingSearchers=X" mean?

Whenever a commit happens in Solr, a new "searcher" (with new caches) is opened, "warmed" up according to your SolrConfigXml settings, and then put in place. The previous searcher is not closed until the "warming" search is ready. If multiple commits happen in rapid succession, then a searcher may still be warming up when another commit comes in. This will cause multiple searchers to be started and warming at the same time, all competing for resources. Only one searcher is ever actively handling queries at a time – all old searchers are thrown away when the latest one finishes warming.

maxWarmingSearchers is a setting in SolrConfigXml that helps you put a safety valve on the number of overlapping warming searchers that can exist at one time. If you see this error it means Solr prevented a new searcher from being opened because there were already X searchers warming up.

The best way to prevent this log message is to reduce the frequency of commit operations. Enough time should pass between commits for each commit to completely finish.

If you encounter this error a lot, you can increase the number for maxWarmingSearchers, but this is usually the wrong thing to do. It requires significant system resources (RAM, CPU, etc...) to do it safely, and it can drastically affect Solr's performance.

If you only encounter this error infrequently because of an unusually high load on the system, you'll probably be OK just ignoring it.

Why doesn't my index directory get smaller (immediately) when i delete documents? force a merge? optimize?

Because of the "inverted index" data structure, deleting documents only annotates them as deleted for the purpose of searching. The space used by those documents will be reclaimed when the segments they are in are merged.

When segments are merged (either because of the Merge Policy as documents are added, or explicitly because of a forced merge or optimize command) then Solr attempts to delete old segment files, but on some filesystems Notably in Microsoft Windows) it is not possible to delete a file while the file is open for reading (Which is usually true since Solr is still serving requests against the old segments until the new Searcher is ready and has it's caches warmed). When this happens, the older segment files are left on disk, and Solr will re-attempt to delete them later the next time a merge happens.

Searching

Why Aren't Scores returned as a percentage? How Do I normalize Scores?

Please see: https://wiki.apache.org/lucene-java/ScoresAsPercentages

How to make the search use AND semantics by default rather than OR?

In `schema.xml`:

```
<solrQueryParser defaultOperator="AND"/>
```

Why does 'foo AND -baz' match docs, but 'foo AND (-bar)' doesn't ?

Boolean queries must have at least one "positive" expression (ie; MUST or SHOULD) in order to match. Solr tries to help with this, and if asked to execute a BooleanQuery that does contains only negatived clauses *at the topmost level*, it adds a match all docs query (ie: *:*)

If the top level BoolenQuery contains somewhere inside of it a nested BooleanQuery which contains only negated clauses, that nested query will not be modified, and it (by definition) an't match any documents – if it is required, that means the outer query will not match.

More Detail:

- https://issues.apache.org/jira/browse/SOLR-80
- https://mail-archives.apache.org/mod_mbox/lucene-solr-user/201006.mbox/%3Calpine.DEB.1.10.1006011609080.29455@radix.cryptio.net%3E

How do I add full-text summaries to my search results?

Basic highlighting/summarization can be added adding `hl=true` to the query parameters. More advanced highlighting is described in HighlightingParameters.

I have set `hl=true` but no summaries are being output

For a field to be summarizable it must be both stored and indexed. Note that this can significantly increase the index size for large fields (e.g. the main content field of a document). Consider storing the field using compression (`compressed=true` in the `schema.xml` `fieldType` definition). Additionally, such field needs to be tokenized.

I want to add basic category counts to my search results

Solr provides support for "facets" out-of-the-box. See SimpleFacetParameters.

How can I figure out why my documents are being ranked the way they are?

Solr's uses Lucene for ranking. A detailed summary of the ranking calculation can be obtained by adding `debugQuery=true` to the query parameter list. The output takes some getting used to if you are not familiar with Lucene's ranking model.

The SolrRelevancyFAQ has more information on understanding why documents rank the way they do.

Why Isn't Sorting Working on my Text Fields?

Lucene Sorting requires that the field you want to sort on be indexed, but it cannot contain more than one "token" per document. Most Analyzers used on Text fields result in more than one token, so the simplest thing to do is to use copyField to index a second version of your field using the StrField class.

If you need to do some processing on the field value using TokenFilters, you can also use the KeywordTokenizer, see the Solr example schema for more information.

My search returns too many / too little / unexpected results, how to debug?

The best way to debug such problems is with the analyzer admin tool, which is at http://localhost:8983/solr/admin/analysis.jsp if using the default configuration.

That page will show you how your field is processed while indexing and while querying, and if a particular query matches.

See also the Solr tutorial and the xml.com article about Solr, listed in the SolrResources.

How can I get ALL the matching documents back? ... How can I return an unlimited number of rows?

This is impractical in most cases. People typically only want to do this when they know they are dealing with an index whose size guarantees the result sets will be always be small enough that they can feasibly be transmitted in a manageable amount – but if that's the case just specify what you consider a "manageable amount" as your `rows` param and get the best of both worlds (all the results when your assumption is right, and a sanity cap on the result size if it turns out your assumptions are wrong)

In cases where you need all the results for external processing, you can either use a cursor, or the export response writer...

- https://cwiki.apache.org/confluence/display/solr/Pagination+of+Results
- https://cwiki.apache.org/confluence/display/solr/Exporting+Result+Sets

Can I use Lucene to access the index generated by SOLR?

Yes, although this is not recommended. Writing to the index is particularly tricky. However, if you do go down this route, there are a couple of things to keep in mind. Be careful that the analysis chain you use in Lucene matches the one used to index the data or you'll get surprising results. Also, be aware that if you open a searcher, you won't see changes that Solr makes to the index unless you reopen the underlying readers.

Is there a limit on the number of keywords for a Solr query?

No. If you make a GET query, through Solr Web interface for example, you are limited to the maximum URL lenght of the browser.

How can I efficently search for all documents that contain a value in fieldX ?

If the number of unique terms in fieldX is bounded and relatively small (ie: a "category" or "state" field) or if fieldX is a "Trie" Numeric field with a small precision step then you will probably find it fast enough to do a simple range query on the field – ie: `fieldX:[* TO *]`. When possible, doing these in cached filter queries (ie: "fq") will also improve performance.

A more efficient method is to also ensure that your index has an additional field which records wether or not each document has a value – ie: `has_fieldX` as a boolean field that can be queried with `has_fieldX:true`, or `num_values_fieldX` that can be queried with `num_values_fieldX:[1 TO *]`. This technique requires you to know in advance that you will want to query on this type of information, so that you can add this extra field to your index, but it can be significantly faster.

Adding a field like `num_values_fieldX` is extremely easy to do automaticly in Solr4.0 by modifying your `<updateRequestProcessorChain>` to include the `CountFieldValuesUpdateProcessorFactory`

Solr Cloud

Can I reuse the same ZooKeeper cluster with other applications, or multiple SolrCloud clusters?

Yes, You can use SolrCloud with any existing ZooKeeper installation running other apps, or a ZooKeeper installation powering an existing SolrCloud cluster (s)

For simplicity, Solr creates nodes at the ZooKeeper root, but a distinct chroot option can be specified for each SolrCloud cluster to isolate them.

Why doesn't SolrCloud automatically create replicas when I add nodes?

There's no way that Solr can guess what the user's intentions are when adding a new node to a SolrCloud cluster.

If Solr automatically creates replicas on a cloud with billions of documents, it might take *hours* for that replication to complete. Users with very large indexes would be VERY irritated if this were to happen automatically.

The new nodes might be intended for an entirely new collection, not new replicas on existing collections. Users who have this intention would also be unhappy if Solr decided to add new replicas.

Even when the intent **is** to add new replicas, Solr has no way of knowing **which** collections should be replicated. On a very large cloud with hundreds of collections, choosing to add a replica to **all** of them could take a very long time and use up all the disk space on the new node.

Additionally, creating replicas uses a lot of disk and network I/O bandwidth. If a node is added during normal hours and replication starts automatically, it might drastically affect query performance.

Performance

How fast is indexing?

Indexing performance varies considerably depending on the size of the documents, the analysis requirements, and CPU and I/O performance of the machine. Rates reported by users vary greatly. Some can index only a few documents per second, others see several thousand per second.

How can indexing be accelerated?

A few ideas:

- Include multiple documents in a single <add> operations. Note: don't put a huge number of documents in each add operation. With very large documents, you may only want to index them ten or twenty at a time. For small documents, between 100 and 1000 is more reasonable.
- Ensure you are not performing <commit/> until you need to see the updated index.
- If you are reindexing every document in your index, completely removing the index first can substantially speed up the required time and disk space.
- Solr can do some, but not all, parts of indexing in parallel. Indexing with multiple client threads can be a boon, particularly if you have multiple CPUs in your Solr server and your analysis requirements are considerable.
- Experiment with different `mergeFactor` and `maxBufferedDocs` settings (see http://www.onjava.com/pub/a/onjava/2003/03/05/lucene.html).

How can I speed up facet counts?

Performance problems can arise when faceting on fields/queries with many unique values. If you are faceting on a tokenized field, consider making it untokenized (field class `solr.StrField`, or using `solr.KeywordTokenizerFactory`).

Also, keep in mind that Solr must construct a filter for every unique value on which you request faceting. This only has to be done once, and the results are stored in the `filterCache`. If you are experiencing slow faceting, check the cache statistics for the `filterCache` in the Solr admin. If there is a large number of cache misses and evictions, try increasing the capacity of the `filterCache`.

What does "PERFORMANCE WARNING: Overlapping onDeckSearchers=X" mean in my logs?

This warning means that at least one searcher hadn't yet finished warming in the background, when a commit was issued and another searcher started warming. This can not only eat up a lot of ram (as multiple on deck searches warm caches simultaneously) but it can can create a feedback cycle, since the more searchers warming in parallel means each searcher might take longer to warm.

Typically the way to avoid this error is to either reduce the frequency of commits, or reduce the amount of warming a searcher does while it's on deck (by reducing the work in newSearcher listeners, and/or reducing the autowarmCount on your caches)

See also the `<maxWarmingSearchers/>` option in SolrConfigXml.

Why is the QTime Solr returns lower then the amount of time I'm measuring in my client?

"QTime" only reflects the amount of time Solr spent processing the request. It does not reflect any time spent reading the request from the client across the network, or writing the response back to the client. (This should hopefully be obvious since the QTime is actually included in body of the response.)

The time spent on this network I/O can be a non-trivial contribution to the the total time as observed from clients, particularly because there are many cases where Solr can stream "stored fields" for the response (ie: requested by the "fl" param) directly from the index as part of the response writing, in which case disk I/O reading those stored field values may contribute to the total time observed by clients outside of the time measured in QTime.

Developing

Where can I find the latest and Greatest Code?

In the Solr Version Control Repository.

Where can I get the javadocs for the classes?

There are official Javadocs of the latest released version available. If you are woking on trunk, please look at nightly Solr Javadocs builds.

How can I help?

Joining and participating in discussion on the developers email list is the best way to get your feet wet with Solr development.

There is also a TaskList containing all of the ideas people have had about ways to improve Solr. Feel free to add your own ideas to this page, or investigate possible implementations of existing ideas. When you are ready, submit a patch with your changes.

How can I submit bug reports, bug fixes or new features?

Bug reports, and patch submissions should be entered in Solr's Bug Tracking Queue.

How do I apply patches from JIRA issues?

Information about testing patches can be found on the How To Contribute wiki page

I can't compile Solr, ant says "JUnit not found" or "Could not create task or type of type: junit"

As of September 21, 2007, JUnit's JAR is now included in Solr's source repository, so there is no need to install it separately to run Solr's unit tests. If ant generates a warning that it doesn't understand the junit task, check that you have an "ant-junit.jar" in your ANT_LIB directory (it should be included when you install apache-ant).

If you are attempting to compile the Solr source tree from prior to September 21, 2007 (including Ant documentation of JUnit]) you will need to include the junit.jar in your ant classpath. Please see the [http://ant.apache.org/manual/OptionalTasks/junit.html for notes about where Ant expects to find the JUnit JAR and Ant task JARs.

How can I start the example application in Debug mode?

You can start the example application in debug mode to debug your java class with your favorite IDE (like eclipse).

```
java -Xdebug -Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=n -jar start.jar
```

Then connect to port 8000 and debug.

Tagging using SOLR

There is a wiki page on some brainstorming on how to implement tagging within Solr [UserTagDesign].

How can I get hold of HttpServletRequest object in custom first-component?

Set the attribute "addHttpRequestToContext" in the "requestParsers" element to "true" in your solrconfig.xml.

Use it in your custom componet like:

```
public class CustomSearchComponent extends SearchComponent {


        @Override
        public void prepare(ResponseBuilder responseBuilder) throws IOException{
                SolrQueryRequest req = responseBuilder.req;
                HttpServletRequest request = (HttpServletRequest)req.getContext().get("httpRequest");
        }
}
```

See https://cwiki.apache.org/confluence/display/solr/RequestDispatcher+in+SolrConfig#RequestDispatcherinSolrConfig-requestParsersElement