

Search Quick Reference

This appendix is a convenient reference for common search related request parameters. It is assumed you already read the related material in the book and are just looking for something to jog your memory.

The content is on the following two pages. You can find an electronic PDF version of this appendix here: <http://www.solrenterprisesearchserver.com>. Having it printed is especially convenient.

For More Information:

www.packtpub.com/apache-solr-3-enterprise-search-server/book

Quick reference

A ***** means the parameter can appear a variable number of times.

Core search (Chapter 4, Searching)

<code>qt</code>	The query type . A named request handler.
<code>q</code>	The query string . Usually, as entered by an end user.
<code>defType=luce</code>	The query parser for <code>q</code> . Recommended: <code>dismax</code> (or <code>edismax</code>).
<code>fq *</code>	A filter query . (p. 120)
<code>start=0</code>	The index into the search results to start returning documents.
<code>rows=10</code>	The number of search result documents to return.
<code>fl=*</code>	The field list to retrieve, comma separated. To get scores: <code>*</code> , <code>score</code>
<code>sort=score desc</code>	The sort order. A comma separated list with <code>asc</code> or <code>desc</code> . (p. 142)
<code>wt=xml</code>	The writer type , that is response format. One of <code>xml</code> , <code>json</code> , <code>python</code> , <code>php</code> , <code>phps</code> , <code>ruby</code> , <code>javabin</code> , <code>csv</code> , <code>xslt</code> (p. 121), <code>velocity</code> (p. 282).

Others: `version=2.2`, `omitHeader=off`, `timeAllowed=-1`

Diagnostic related (Chapter 4, Searching)

`indent=off`, `debugQuery=off`, `explainOther` (a query for one doc), `debug.explain.structured=off`, `echoParams=explicit` (`none` | `explicit` | `all`), `echoHandler=off`

Tip: Use `wt=xslt&tr=example.xsl&debugQuery=true&fl=*,score`

Lucene query parser (Chapter 4, Searching)

<code>df</code>	The default field to search.
<code>q.op=OR</code>	The default query operator . One of <code>AND</code> , <code>OR</code> .

Dismax query parser (Chapter 4, Searching and Chapter 5, Search Relevancy)

<code>q.alt</code>	An alternate query to run when <code>q</code> is absent. Recommend: <code>*:*</code> (all docs)
<code>qf</code>	The query fields , including optional boosts. Ex: <code>id^5.0 name^2.0 body</code>
<code>mm=100%</code>	The min-should-match specification. To change to all-optional, use <code>0%</code>
<code>qs=0</code>	The query slop for phrases <i>explicitly</i> in the query string.
<code>pf</code>	The phrase fields for automatic phrase boosting. Same <code>qf</code> syntax.
<code>ps=0</code>	The phrase slop for <code>pf</code> .
<code>tie=0</code>	The score tie-breaker. Recommend: <code>0.1</code>
<code>bq *</code>	A boost query . The boost is added.

<code>bf *</code>	A boost function. The boost is added.
<code>boost *</code>	A boost function. The boost is multiplied. <i>edismax only</i>

Other edismax additions: `pf2`, `pf3`, `stopwords=on`, `lowercaseOperators=on`

Lucene Query Syntax (Chapter 4, Searching)

Boolean operators: AND, OR, NOT, &&, ||. Leading + or -.

Ex: `{!lucene df=title q.op=$myop} "phrase query slop"~2 w?ldcard* fuzzy~0.7 - (updatedAt:[* TO NOW/DAY-2YEAR] +boostMe^5)`

Faceting (Chapter 6, Faceting)

Example of a field specific parameter: `f.myfieldname.facet.mincount=1`

Field value faceting: `facet=off`, `facet.field *`, `facet.sort=count` (`count` | `lex`), `facet.limit=100`, `facet.offset=0`, `facet.mincount=0`, `facet.missing=off`, `facet.prefix`, `facet.method` (`enum` | `fc`)

Range faceting: `facet=off`, `facet.range *` (field name), `facet.range.start`, `facet.range.end`, `facet.range.gap` (ex: `+1DAY`), `facet.range.hardend=off`, `facet.range.other=off`, `facet.range.include=lower *` (`lower`, `upper`, `edge`, `outer`, `all`)

Facet queries: `facet=off`, `facet.query *`

Exclude a filter: `fq={!tag=r_type}r_type:Album&facet.field={!ex=r_type}r_type`

Highlighting (Chapter 7, Search Components, p. 202)

Common parameters: `hl=off`, `hl.fl`, `hl.requireFieldMatch=off`, `hl.usePhraseHighlighter=off` (recommend on), `hl.highlightMultiTerm=off`, `hl.snippets=1`, `hl.fragment=100`, `hl.mergeContiguous=off`

Spell checking (Chapter 7, Search Components, p. 215)

Parameters: `spellcheck=off`, `spellcheck.dictionary=default`, `spellcheck.q` (alternative to `q`), `spellcheck.count=1`, `spellcheck.onlyMorePopular=off`, `spellcheck.extendedResults=off`, `spellcheck.collate=off`, `spellcheck.maxCollations=1`, `spellcheck.maxCollationTries=0`, `spellcheck.maxCollationEvaluations=10000`, `spellcheck.collateExtendedResults=off`

Miscellaneous non-search

Commit: `/update?commit=true` (`optimize=true` to optimize)

Delete: `/update?stream.body=<delete><query>*:*/query></delete>`

Reload config: `/admin/cores?action=RELOAD&core=mycorename`