[Apache Solr](#) >

Solr: Query parameters

Search criteria related parameters

q: This refers to the user query or just query for short. This typically originates directly from user input. The query syntax is determined by the defType parameter.

defType: This is a reference to the query parser for the user query in q. The default is lucene. You'll most likely use dismax or edismax as that gives more google search like feature with relaxes syntaxes.

fq: This is a filter query that limits the scope of the user query, similar to a WHERE clause in SQL. Unlike the q parameter, it has no effect on scoring. This parameter can be repeated as desired. Filtering has been described later in the chapter.

Result pagination related parameters

start (default: 0): This is the zero-based index of the first document to be returned from the result set. In other words, this is the number of documents to skip from the beginning of the search results. If this number exceeds the result count, then it will simply return no documents; but this is not considered an error.

rows (default: 10): This is the number of documents to be returned in the response XML, starting at index start. Fewer rows will be returned if there aren't enough matching documents. This number is basically the number of results displayed at a time on your search user interface.

Output-related parameters

fl: This parameter accepts a comma- and/or space-delimited list of values that determine which fields will be present in the response documents. This parameter can be specified multiple times.

sort: This refers to a comma-separated field listing to sort on, with a directionality specifier of asc or desc after each field; for example: r_name asc, score desc. The default is score desc. You can also sort by functions.

wt: This is the response format, also known as writer type or query response writer, defined in solrconfig.xml.Available formats - xml (the default and aliased to standard), json, python, php, phps, ruby, javabin, csv, xslt, velocity.

version: This refers to the requested version of Solr's response structure, if different than the default. Solr's response format hasn't changed in years. However, if Solr's response structure changes, then it will do so under a new version. By using this in the request from client code, a best practice, you reduce the chances of your client code breaking if Solr is updated.

More on fl parameters

The fl parameter accepts a wide range of value types, all of which can be freely mixed together in any order:

Field names: These are simply document field names. Fields added to the fl parameter cause the same fields to be present in the response documents; for example, fl=a_name.

Functions: Any valid Solr function query can be included as a document field value (see the next chapter for more on these functions); for example, fl=sum(1,2,sum(3,4)).

Aliases: Fields can be renamed (aliased) using the fl=new_name:original_name syntax. The result of a function call can also be aliased with fl=ten:product(2,5).
Score: The score for each document can be included in the response by adding score to the fl parameter.

Glob: Use * to refer to all fields and/or partially matching field names. For example, if you want only fields that start with a_, you would use fl=a_*.

Document transformers in fl

explain: This embeds explain information for each document. This transformer accepts an optional style argument set to one of these values: nl, text, or html. fl=*,[explain style=nl]

value: This adds static values to each document. This transformer has one required parameter, v, which sets the value of the field. An optional type parameter t can be set to one of these values: int, double, float, and date. An example is [value v=1 t=double].

Diagnostic parameters

indent: This is a Boolean option that will indent the output to make it easier to read.

debugQuery: If true, then following the search results is <lst name="debug"> with diagnostic information. It contains voluminous information about the parsed query string, how the scores were computed, and timings for all of the Solr components to perform their part of the processing such as faceting. You may need to use the View Source function of your browser to preserve the formatting used in the score computation section.

explainOther: If you want to determine why a particular document wasn't matched by the query or why it wasn't scored highly enough, then you can set a query for this parameter, such as id:"Release:12345", and output of the debugQuery will be sure to include the first document matching this query in its output.

debug: This is a parameter to specify individual debugging features—add a debug parameter pair for each of these values as desired:

 query: Returns information about how the query was parsed

 results: Returns scoring information for each matching document

 timing: Returns component timing information

 true: Equivalent to debugQuery=true

 echoHandler: If true, then this emits the Java class name identifying the Solr request handler.

 echoParams: This controls whether or not query parameters are returned in the response header, as seen verbatim earlier. This is used to debug URL encoding issues, or to verify the complete set of parameters in effect—those present in the request (the URL plus HTTP post data) and those defined in the request handler. Specifying none disables this, which is appropriate for production real-world use. The default value is explain, which causes Solr to include only the parameters present in the request. Finally, you can use all to include those parameters configured in the request handler in addition to those in the URL.

 debug.explain.structured: When true, the result of the score explanation is returned as structured data.

Note:
For the Boolean parameters, a true value can be any one of true, on, or yes. False values can be any of false, off, and no.