TermsComponent

Introduction

⚠️ Solr1.4

The TermsComponent SearchComponent is a simple component that provides access to the indexed terms in a field and the number of documents that match each term. This can be useful for doing auto-suggest or other things that operate at the term level instead of the search or document level. Retrieving terms in index order is very fast since the implementation directly uses Lucene's TermEnum to iterate over the term dictionary.

In a sense, this component provides fast field faceting over the whole index (not restricted by the base query or any filters). The doc frequencies returned are the number of documents that match the term, including any documents that have been marked for deletion but not yet removed from the index.

How it Works

To use the TermsComponent, users can pass in a variety of options to control what terms are returned. The supported parameters are available in the class http://lucene.apache.org/solr/api/org/apache/solr/common/params/TermsParams.html

These params are:

- terms={true|false} - Turn on the TermsComponent
- terms.fl={FIELD NAME} - Required. The name of the field to get the terms from. May be specified multiple times as terms.fl=field1&terms.fl=field2...
- terms.lower={The lower bound term} - Optional. The term to start at. If not specified, the empty string is used, meaning start at the beginning of the field.
- terms.lower.incl={true|false} - Optional. Include the lower bound term in the result set. Default is true.
- terms.mincount=<Integer> - Optional. The minimum doc frequency to return in order to be included. Results are **inclusive** of the mincount (i.e. >= mincount)
- terms.maxcount=<Integer> - Optional. The maximum doc frequency. Default is -1 to have no upper bound. Results are **inclusive** of the maxcount (i.e. <= maxcount)
- terms.prefix={String} - Optional. Restrict matches to terms that start with the prefix.
- terms.regex={String} - Optional. Restrict matches to terms that match the regular expression. ⚠️ Solr3.1
- terms.regex.flag={case_insensitive|comments|multiline|literal|dotall|unicode_case|canon_eq|unix_lines} - Optional. Flags to be used when evaluating the regular expression defined in the "terms.regex" parameter (see http://java.sun.com/j2se/1.5.0/docs/api/java/util/regex/Pattern.html#compile%28java.lang.String,%20int%29 fore more details). This parameter can be defined multiple times (each time with different flag) ⚠️ Solr3.1
- terms.limit={integer} - The maximum number of terms to return. The default is 10. If < 0, then include all terms.
- terms.upper={The upper bound term} - The term to stop at. Either upper or terms.limit must be set.
- terms.upper.incl={true|false} - Include the upper bound term in the result set. Default is false.
- terms.raw={true|false} - If true, return the raw characters of the indexed term, regardless of if it is human readable. For instance, the indexed form of numeric numbers is not human readable. The default is false.
- terms.sort={count|index} - If count, sorts the terms by the term frequency (highest count first). If index, returns the terms in index order. Default is to sort by count.

The output is a list of the terms and their document frequency values.

Distributed Search Support

⚠️ Solr3.1

TermsComponent now supports distributed setups. Assuming that you are using the "/terms" request handler, you should specify the following two parameters to make it work in a distributed setup:

- "shards" - See DistributedSearch
- "shards.qt" - Signals Solr that requests to shards should be sent to a request handler given by this parameter

Examples

The following examples use the Solr tutorial example located in the <Solr>/example directory.

Simple

http://localhost:8983/solr/terms?terms.fl=name&terms.sort=index

Get back the first ten terms in the name field.

Results:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>

<lst name="responseHeader">
 <int name="status">0</int>
 <int name="QTime">1</int>
</lst>
<lst name="terms">
 <lst name="name">
  <int name="0">1</int>
  <int name="1">6</int>
  <int name="11">1</int>
  <int name="120">1</int>
  <int name="133">1</int>
  <int name="184">6</int>
  <int name="19">1</int>
  <int name="1900">1</int>
  <int name="2">4</int>
  <int name="20">1</int>
 </lst>
</lst>
</response>
```

Specifying Lower Bound

http://localhost:8983/solr/terms?terms.fl=name&terms.lower=a&terms.sort=index

Result:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>

<lst name="responseHeader">
 <int name="status">0</int>
 <int name="QTime">0</int>
</lst>
<lst name="terms">
 <lst name="name">
  <int name="a">2</int>
  <int name="adata">2</int>
  <int name="all">1</int>
  <int name="allinone">1</int>
  <int name="apple">1</int>
  <int name="asus">1</int>
  <int name="ata">1</int>
  <int name="ati">1</int>
  <int name="b">1</int>
  <int name="belkin">1</int>
 </lst>
</lst>
</response>
```

Use in Auto-Complete

See also Suggester, which can be a better solution in many scenarios.

To use in auto-complete, add what the user has typed as a prefix:

http://localhost:8983/solr/terms?terms.fl=name&terms.prefix=at

Result:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>

<lst name="responseHeader">
 <int name="status">0</int>
 <int name="QTime">120</int>
</lst>
<lst name="terms">
 <lst name="name">
  <int name="ata">5</int>
  <int name="ati">5</int>
 </lst>
</lst>
</response>
```

You can use the JSON response format, along with omitHeader=true to omit responseHeader for an even smaller response:

http://localhost:8983/solr/terms?terms.fl=name&terms.prefix=at&wt=json&omitHeader=true

Result:

```
{"terms":{"name":["ata",1,"ati",1]}}
```

NOTE: there was a bug in Solr 1.4 that caused the "terms" map to appear as a list.

Case insensitive Auto-Complete

If the analysis if the field preserves case, one can still get case insensitive auto-complete by using the regexp support together with the "case_insensitive" regex flag.

http://localhost:8983/solr/terms?terms.fl=manu_exact&terms.regex=at.*&terms.regex.flag=case_insensitive

Result:

```
{
  "responseHeader":{
    "status":0,
    "QTime":0},
  "terms":{
    "manu_exact":[
      "ATI Technologies",1]}}
```