SearchComponent

Search Components

⚠️ Solr1.3 –
Search components enable a SearchHandler to chain together reusable pieces of functionality to create custom search handlers without writing code.

There are currently several "default" SearchComponents which now comprise the default SearchHandler. They are:

- query - QueryComponent
- facet - FacetComponent
- mlt - MoreLikeThis
- highlight - Highlighting
- stats - Statistics ⚠️ Solr1.4
- debug - Debug

These examples are enabled by default, but can be overridden by writing your own class that extends SearchComponent and then declare it in the solrconfig.xml with the appropriate name. For instance, to override the "query" component, one would declare:

```
<searchComponent name="query"     class="my.app.MyQueryComponent" />
```

Other useful components are:

- SpellCheckComponent – Exposes advanced support for spell checking such as multiple (per-field) dictionaries, loading dictionaries from files, lucene indices and Solr fields, support for analyzers, collation, query parsing and pluggable spell checker implementations. The default implementation provided with Solr uses the Lucene contrib SpellChecker. This component can also provide an autocomplete functionality when used in combination with a Suggester instead of SpellChecker.
- QueryElevationComponent – Used to "elevate" results based on editorial decisions, not relevance.
- TermVectorComponent – Retrieve Term Vector information. See https://issues.apache.org/jira/browse/SOLR-651 ⚠️ Solr1.4
- StatsComponent – Get Numeric field statistics. See https://issues.apache.org/jira/browse/SOLR-680 ⚠️ Solr1.4
- ClusteringComponent – Cluster results and documents. See https://issues.apache.org/jira/browse/SOLR-769 ⚠️ Solr1.4
- TermsComponent – Get access to Lucene's TermEnum capability, providing access to the Term dictionary. See https://issues.apache.org/jira/browse/SOLR-877 ⚠️ Solr1.4
- BloomIndexComponent – Quickly check for the existence of a term using a Bloom Filter. See https://issues.apache.org/jira/browse/SOLR-1375 ⚠️ Solr4.0

Components can also be initialized with NamedList params. For example, the Query Elevator (editorial boosting) can be declared with:

```
<searchComponent name="elevator" class="org.apache.solr.handler.component.QueryElevationComponent" >
    <!-- pick a fieldType to analyze queries -->
    <str name="queryFieldType">string</str>
    <str name="config-file">elevate.xml</str>
  </searchComponent>
```

Components can be reused by multiple instances of SearchHandler – either by prepending (first-components), appending (last-components), or replacing (components) the default list...

```
<requestHandler name="/elevate" class="solr.SearchHandler">
    <!-- add my elevator component to the end of the default list -->
    <arr name="last-components">
      <str>elevator</str>
    </arr>
</requestHandler>
<requestHandler name="/simple" class="solr.SearchHandler">
    <!-- i don't want many of the defaults -->
    <arr name="components">
      <str>query</str>
      <str>debug</str>
    </arr>
</requestHandler>
```

---

see: http://www.nabble.com/search-components-%28plugins%29-tf3898040.html#a11050274

It was first introduce as part of SOLR-281

<<FullSearch(title:"SearchComponent/")>>