

schema.xml

schema.xml is usually the first file you configure when setting up a new Solr installation.

The schema declares:

- what kinds of fields there are
- which field should be used as the unique/primary key
- which fields are required
- how to index and search each field

The XML consists of a number of parts. We'll look at these in turn:

- **Field Types**
- **Fields**
- **Misc**

Field Types

```
<types>
  <fieldType name="int" class="solr.TrieIntField"precisionStep="0" omitNorms="true"positionIncrementGap="0"/>
  ...
</types>
```

The example Solr schema.xml comes with a number of pre-defined field types, and they're quite well-documented. You can also use them as templates for creating new field types.

The commonly used ones are:

text

A generically useful text field. Its described in the documentation as:

A text field that uses WordDelimiterFilter to enable splitting and matching of words on case-change, alpha numeric boundaries, and non-alphanumeric chars, so that a query of "wifi" or "wi fi" could match a document containing "Wi-Fi". Synonyms and stopwords are customized by external files, and stemming is enabled.

string

Useful when you have a text field which you don't want tokenized, like IDs. Its described in the documentation as:

The StrField type is not analyzed, but indexed/stored verbatim. - StrField and TextField support an optional compressThreshold which limits compression (if enabled in the derived fields) to values which exceed a certain size (in characters).

date

Useful for dates. Its described in the documentation as:

The format for this date field is of the form 1995-12-31T23:59:59Z, and is a more restricted form of the canonical representation of dateTime <http://www.w3.org/TR/xmlschema-2/#dateTime>

float and int

Self-explanatory.

You can find **a list of Java classes which implement FieldType here**.

The Solr Wiki also has some information on **field types**.

Fields

```
<fields>
  <field name="id" type="string" indexed="true"stored="true" required="true" />
  <field name="name" type="textgen"indexed="true" stored="true"/>
  ...
</fields>
```

The documentation provides a list of valid attributes:

- **name**: mandatory - the name for the field
- **type**: mandatory - the name of a previously defined type from the <types> section
- **indexed**: true if this field should be indexed (searchable or sortable)
- **stored**: true if this field should be retrievable
- **compressed**: [false] if this field should be stored using gzip compression (this will only apply if the field type is compressable; among the standard field types, only TextField and StrField are)
- **multiValued**: true if this field may contain multiple values per document
- **omitNorms**: (expert) set to true to omit the norms associated with this field (this disables length normalization and index-time boosting for the field, and saves some memory). Only full-text fields or fields that need an index-time boost need norms.
- **termVectors**: [false] set to true to store the term vector for a given field. When using MoreLikeThis, fields used for similarity should be stored for best performance.
- **termPositions**: Store position information with the term vector. This will increase storage costs.
- **termOffsets**: Store offset information with the term vector. This will increase storage costs.
- **default**: a value that should be used if no value is specified when adding a document.

The Solr Wiki has more information on [fields](#) like dynamic fields etc.

uniqueKey

```
<uniqueKey>id</uniqueKey>
```

Equivalent to the primary key of the document.

Field to use to determine and enforce document uniqueness. Unless this field is marked with `required="false"`, it will be a required field

defaultSearchField

```
<defaultSearchField>aggregate_text</defaultSearchField>
```

Field for the QueryParser to use when an explicit fieldname is absent

solrQueryParser

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

Used for determining if multiple terms are ANDed or ORed together by default.

SolrQueryParser configuration: `defaultOperator="AND|OR"`

For example, with the following query

```
quick brown fox
```

a setting of

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

will produce the following Solr boolean query

```
quick AND brown AND fox
```

Note: We've only covered the most commonly-used configuration elements. The Solr Wiki has an [extensive list of configuration elements in schema.xml](#).