Solr notes

(Reference: <https://examples.javacodegeeks.com/enterprise-java/apache-solr/apache-solr-tutorial-beginners/> )

Ahm MMR module adopted Camel+Cassandra+solr tech stack. Ahm messages come from ETAMS queue routed to the message handler by apache Camel light weight ESB framework. The message handler inserted it into event table and forward message into java PriorityBlockingQueue ordered by priority. The message is further proceeded by messageConsumer parsing it from raw data to different tables in Cassandra. At the meantime, many quartz schedulers are triggered in various frequencies. One of job is index data from Cassandra to solr document. This document is quick start of Solr. It can help you to quickly catch the key concepts of Solr.

. Install Solr

Different versions have different folder structures. It is one of the learning curve of solr. In Ahm, version 4.9.0 was adopted. The latest version is 5.x. It is a zip file. Unzip it done.

. Create your own core/collection and schema

When the Solr server is started in Standalone mode the configuration is called core and when it is started in SolrCloud mode the configuration is called Collection.

Solr 4.x create core

(Reference: <http://wiki.apache.org/solr/CoreAdmin#CREATE> )

The home folder of multiple core is solr-4.9.0/example/solr. If we want to create our own core, the following steps must be done.

1. Create the mycore folder(the folderName is the core instance name we want to create)
2. Go to sample folder of collection1
3. Copy conf folder under collection1 folder to mycore folder
4. Run Java –jar start.jar to startup solr server
5. Run <http://localhost:8983/solr/admin/cores?action=CREATE&name=mycore&config=solrconfig.xml&schema=schema.xml&dataDir=data>

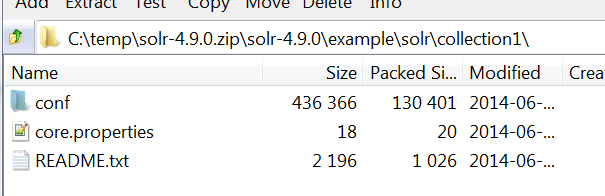
At the browser

OR

Go to <http://localhost:8983/solr> --> core admin 🡪 click add core button ->fill the form

Done!!

Step 5 is just register the new instance into solr server.



Solr 5.x Create core:

The home folder of multiple cores is server\solr in Solr 5.x. Creating core in 5.x+ is much easier than 4.x. it doesn’t need prepare for the coreInstance folder and configure files. We can directly run solr command line ‘bin/solr –c mycorename’

First, The Solr create command has the following options:

* -c <name> – Name of the core or collection to create (required).
* -d <confdir> – The configuration directory, useful in the SolrCloud mode.
* -n <configName> – The configuration name. This defaults to the same name as the core or collection.
* -p <port> – Port of a local Solr instance to send the create command to; by default the script tries to detect the port by looking for running Solr instances.
* -s <shards> – Number of shards to split a collection into, default is 1.
* -rf <replicas> – Number of copies of each document in the collection. The default is 1.

Modify schema.xml

The schema definition must come from the indexed document. In the schema.xml file, the field type is the most important and also the most difficult portion. We have to recognize Solr having analysis step during the index process. Solr will tokenize each character in the stream in order to be able to search on it. This is reason Solr has ‘index’ and ‘store’ concepts in schema file. Index=true means the field is searchable, but doesn’t mean it can be displayed correctly since solr only has the tokenized data. If we like to display the data, we have to set ‘store=true’. ‘store=true’ means the data will be stored as is. For the basic concept, we have to understand the different from string and text. ‘string’ type means the data will not be tokenized char by char. Therefore, we can only search it by entire string. Type ‘text’ will be tokenized, so we can search it by wildcard. Other types we can read the solr document.

dynamicField

<dynamicField name="\*\_i" type="int" indexed="true" stored="true"/>

It is similar to field, but it can contains wildcard. It saves us time to define a group of fields.

Index data

Once we have schema defined and has corresponding documents ready, we can upload the documents into solr. This upload process is called index since solr will not only just store the data into solr server, but also analyze the document per the schema we defined earlier.

Solr 5+ provides a command line tool called post.

java -jar post.jar -h

The usage format in general is as follows  
Usage: java [SystemProperties] -jar post.jar [-h|-] [<file|folder|url|arg>  
[<file|folder|url|arg>...]]

As we said earlier, we will index the data present in the “books.csv” file shipped with Solr installation. We will navigate to the solr-5.0.0\example\exampledocs in the command prompt and issue the following command.

java -Dtype=text/csv -Durl=http://localhost:8983/solr/jcg/update -jar post.jar  books.csv

The SystemProperties used here are:

* -Dtype – the type of the data file.
* -Durl – URL for the jcg core.

Access data

Open the following URL in a browser.

http://localhost:8983/solr/mycorename/select?q=name:"A Clash of Kings"

the detail queries see solr doc.