**Apache Solr 5.5.0**

**About Solr**

Solr is highly reliable, scalable and fault tolerant, providing distributed indexing, replication and load-balanced querying, automated failover and recovery, centralized configuration and more. Solr powers the search and navigation features of many of the world's largest internet sites.

**Features**

* **Advanced Full-Text Search Capabilities**

Powered by Lucene™, Solr enables powerful matching capabilities including phrases, wildcards, joins, grouping and much more across any data type

* **Optimized for High Volume Traffic**

Solr is proven at extremely large scales the world over

* **Standards Based Open Interfaces - XML, JSON and HTTP**

Solr uses the tools you use to make application building a snap

* **Comprehensive Administration Interfaces**

Solr ships with a built-in, responsive administrative user interface to make it easy to control your Solr instances

* **Easy Monitoring**

Need more insight into your instances? Solr publishes loads of metric data via JMX

* **Highly Scalable and Fault Tolerant**

Built on the battle-tested Apache Zookeeper, Solr makes it easy to scale up and down. Solr bakes in replication, distribution, rebalancing and fault tolerance out of the box.

* **Flexible and Adaptable with easy configuration**

Solr's is designed to adapt to your needs all while simplifying configuration

* **Near Real-Time Indexing**

Want to see your updates now? Solr takes advantage of Lucene's Near Real-Time Indexing capabilities to make sure you see your content when you want to see it

* **Extensible Plugin Architecture**

Solr publishes many well-defined extension points that make it easy to plugin both index and query time plugins. Of course, since it is Apache-licensed open source, you can change any code you want!

**Why solr**

Apache Solr is a powerful search server, which supports REST like API. Solr is powered by Lucene which enables powerful matching capabilities like phrases, wildcards, joins, grouping and many more across various data types. It is highly optimized for high traffic using Apache Zookeeper. Apache Solr comes with a wide set of features and we have listed a subset of  high impact features.

* Advanced Full-Text search capabilities.
* Standards based on Open Interfaces – XML, JSON and Http.
* Highly scalable and fault tolerant.
* Supports both Schema and Schemaless configuration.
* Faceted Search and Filtering.
* Support major languages like English, German, Chinese, Japanese, French and many more
* Rich Document Parsing.

**Installation :**

To begin with lets download the latest version of Apache Solr from the following location:

http://lucene.apache.org/solr/downloads.html

As of this writing, the stable version available is 5.0.0. Apache Solr has gone through various changes from 4.x.x to 5.0.0, so if you have different version of Solr you need to download the 5.x.x. version .

Once the Solr zip file is downloaded unzip it into a folder.

**Start Solr Server**

We can start the server using the command line script. Lets go to the bin directory from the command prompt and issue the following command

**solr start**

**Create Core**

**Solr create –c “<core name>”**

**Configuration files :**

**Schema.xml**

We have to ensure that the following line is not commented in solrConfig file <schemaFactory class="ClassicIndexSchemaFactory"/> else Solr will load the schema from the resource named in 'managedSchemaResourceName', rather than from schema.xml Note that the managed schema resource CANNOT be named schema.xml. If the manage schema does not exist, Solr will create it after reading schema.xml, then rename 'schema.xml' to 'schema.xml.bak'.

needs to specify the fields on which we are going to search.

Example :

<dynamicField name="\*\_name" type="text\_general" multiValued="false" indexed="true" stored="true" /> - required only for DB

<uniqueKey>id</uniqueKey>

- id field is important for indexing without that indexing will not happen.

<field name="empName" type="text\_general" indexed="true" stored="true"/>

- we have to give the field names which we gave it in db-data-config.xml. Then only solr will produce the result.

<field name="emp\_add" type="text\_general" indexed="true" stored="true"/>

**solrConfig.xml**

We have to add request handler for DB. Not needed for Document search

<lib dir="${solr.install.dir:../../../..}/contrib/dataimporthandler/lib/" regex=".\*\.jar" />

<lib dir="${solr.install.dir:../../../..}/dist/" regex="solr-dataimporthandler-\d.\*\.jar" />

<lib dir="${solr.install.dir:../../../..}/server/solr/Sample/lib/" regex="postgresql-\d.\*\.jar" />

<requestHandler name="/dataimport" class="org.apache.solr.handler.dataimport.DataImportHandler">

<lst name="defaults">

<str name="config">db-data-config.xml</str>

</lst>

</requestHandler>

**db-data-config.xml(Required only for DB)**

**<dataConfig>**

**<dataSource type="JdbcDataSource"**

**driver="org.postgresql.Driver"**

**url="jdbc:postgresql://localhost:5432/postgres"**

**user="postgres"**

**password="POSTGRES" />**

<document name="employees">

<entity name="emp\_dtl" query="select \* FROM emp\_dtl">

<field column="emp\_no" name="id"/>

<field column="emp\_nm" name="empName"/>

<field column="emp\_add" name="empAdd"/>

</entity>

</document>

</dataConfig>