**Cloudera Search Full Text Search with Solr**

**HBase Batch Indexer and Search**

**Author: Michelle Zhang**

**Company: Cloudera**

**Date: 10/28/2014**

# Install Cloudera Search Related Components in Cluster

## Install Solr Related Components

After you installed CM, you can use CM “Add a Service” entry to automatically install Solr Related Components.

Base Components: ZooKeeper, HDFS, YARN(MR2 included), HBase

Cloudera Search Need Components: Solr, Flume, Key-Value Store Indexer, Hue

## Start Solr Related Components’ Services

Start Solr related components’ services in sequence with CM:

CM 🡪 Home 🡪 Start Services(Solr, Flume, Key-Value Store Indexer, Hue)

Take CDH 5.2 cluster as an example, and set $HOME=/root/solr

# Create HBase Table

Create HBase Table and input some data into the table.

# create 'record', {NAME => 'data'}

# put 'record', 'row1', 'data', 'value'

# put 'record', 'row2', 'data', 'value2'

# Create Collection for HDFS Batch Indexer and Search

## Generate Configuration for Collection

Generate the configuration files for the collection:

# solrctl instancedir --generate /root/solr/hbase-collection1

## Prepare Schema.xml for collection

Edit /root/solr/hbase-collection1/conf/schema.xml:

<fields>

<field name="data" type="text\_general" indexed="true" stored="true" required="true" multiValued="false" />

<field name="id" type="string" indexed="true" stored="true" required="true" />

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

<field name="\_version\_" type="long" indexed="true" stored="true"/>

</fields>

<uniqueKey>id</uniqueKey>

<copyField source="data" dest="text"/>

<copyField source="id" dest="text"/>

## Upload Instance Directory to ZooKeeper

# solrctl instancedir --create hbase-collection1 /root/solr/hbase-collection1

## Create New Collection in HDFS

# solrctl collection --create hbase-collection1

Then you can verify whether the collection is live in Solr Server Web UI.

For example: <http://ec2-54-69-194-12.us-west-2.compute.amazonaws.com:8983/solr/#/~cloud>



## Create Lily HBase Indexer Configuration

# cat /root/solr/morphline-hbase-mapper.xml

<?xml version="1.0"?>

<indexer table="record" mapper="com.ngdata.hbaseindexer.morphline.MorphlineResultToSolrMapper">

<!-- The relative or absolute path on the local file system to the morphline configuration file. -->

<!-- Use relative path "morphlines.conf" for morphlines managed by Cloudera Manager -->

<param name="morphlineFile" value**="/etc/hbase-solr/conf/morphlines.conf**"/>

<!-- The optional morphlineId identifies a morphline if there are multiple morphlines in morphlines.conf -->

<!-- <param name="morphlineId" value="morphline1"/> -->

</indexer>

## Create Morphline

It related to Lily HBase -> Configurations->Service-Wide->Morphlines. But no need to configure it when running batch indexing.

When running batch indexing, just need to edit following file.

# cat /etc/hbase-solr/conf/morphlines.conf

SOLR\_LOCATOR : {

# Name of solr collection

collection : hbase-collection1

# ZooKeeper ensemble

zkHost : "$ZK\_HOST"

}

morphlines : [

{

id : morphline1

importCommands : ["org.kitesdk.\*\*", "com.ngdata.\*\*"]

commands : [

{

**extractHBaseCells** {

mappings : [

{

inputColumn : "data:\*"

outputField : "data"

type : string

source : value

}

]

}

}

{ logDebug { format : "output record: {}", args : ["@{}"] } }

]

}

]

# HBase Batch Indexer

## Run HBaseMapReduceIndexerTool to index data

# hadoop --config /etc/hadoop/conf.cloudera.yarn \

jar /opt/cloudera/parcels/CDH/lib/hbase-solr/tools/hbase-indexer-mr-1.5-cdh5.2.0-job.jar \

--conf /etc/hbase/conf.cloudera.hbase/hbase-site.xml \

-D 'mapred.child.java.opts=-Xmx500m' \

--hbase-indexer-file /root/solr/morphline-hbase-mapper.xml \

--zk-host **$ZKHOME**/solr --collection **hbase-collection1** \

**--go-live** --log4j /etc/hbase/conf.cloudera.hbase/log4j.properties

## Check ResourceManager about Indexer Job

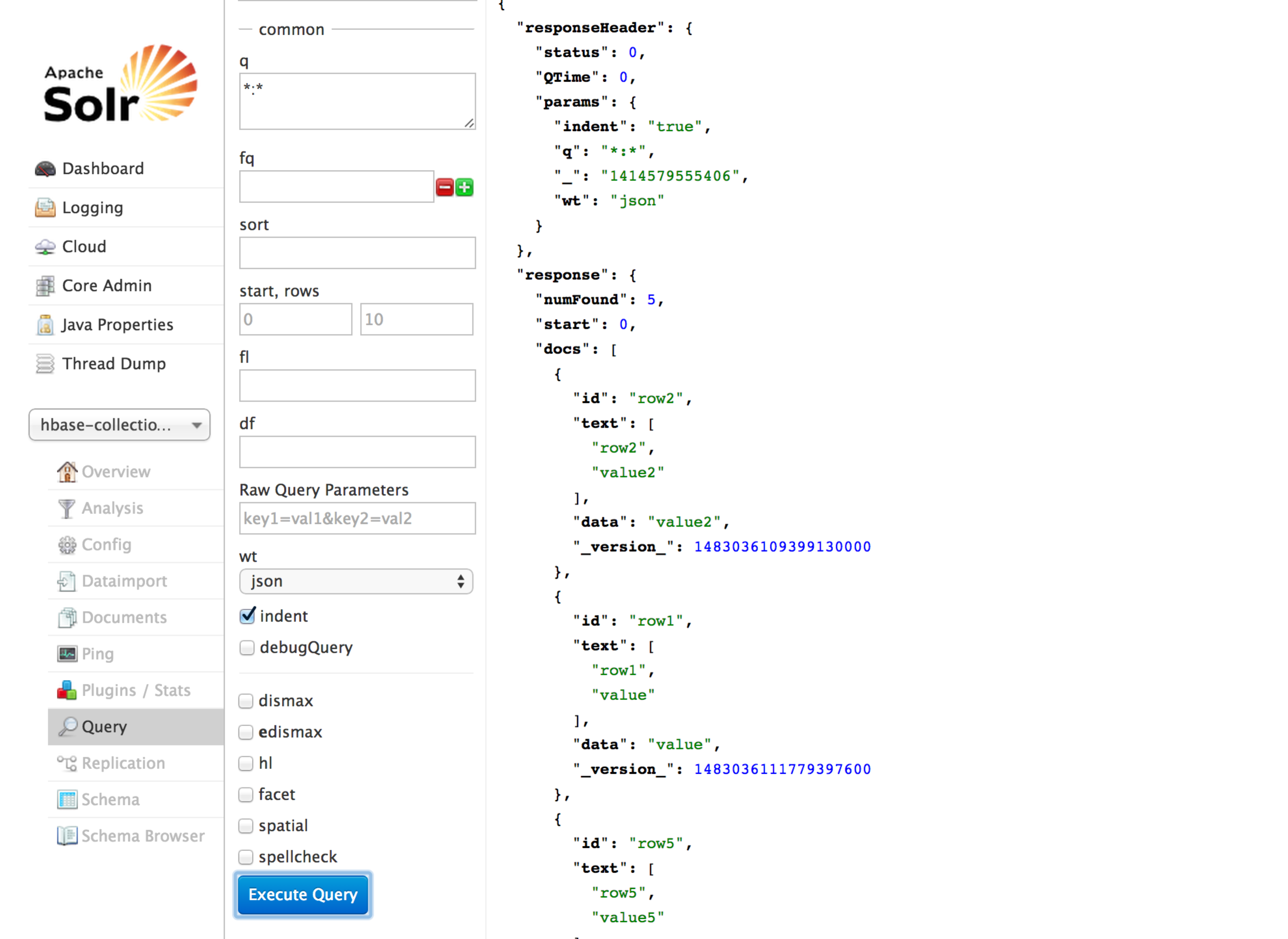
There will be two MapReduce jobs named as “HBaseMapReduceIndexerTool/HBaseIndexerMapper” and “org.apache.solr.hadoop.ForkedMapReduceIndexerTool/ForkedTreeMergeMapper” running for indexer.

# Cloudera Search Result in Solr and Hue UI

## Check Solr Queries

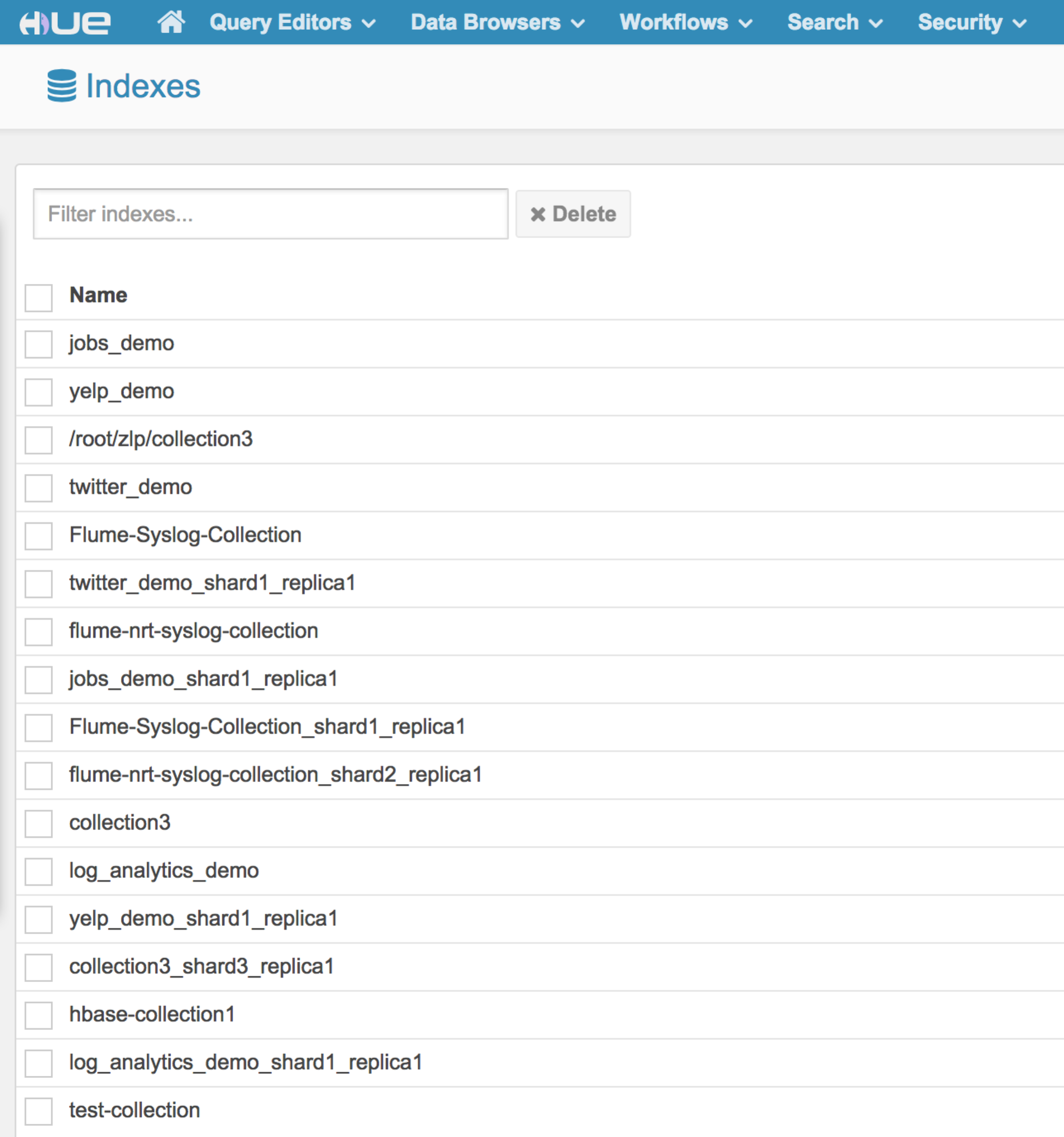
You can check Solr Server Web UI to query the results

<http://ec2-54-69-242-17.us-west-2.compute.amazonaws.com:8983/solr/#/hbase-collection1_shard1_replica1/query>



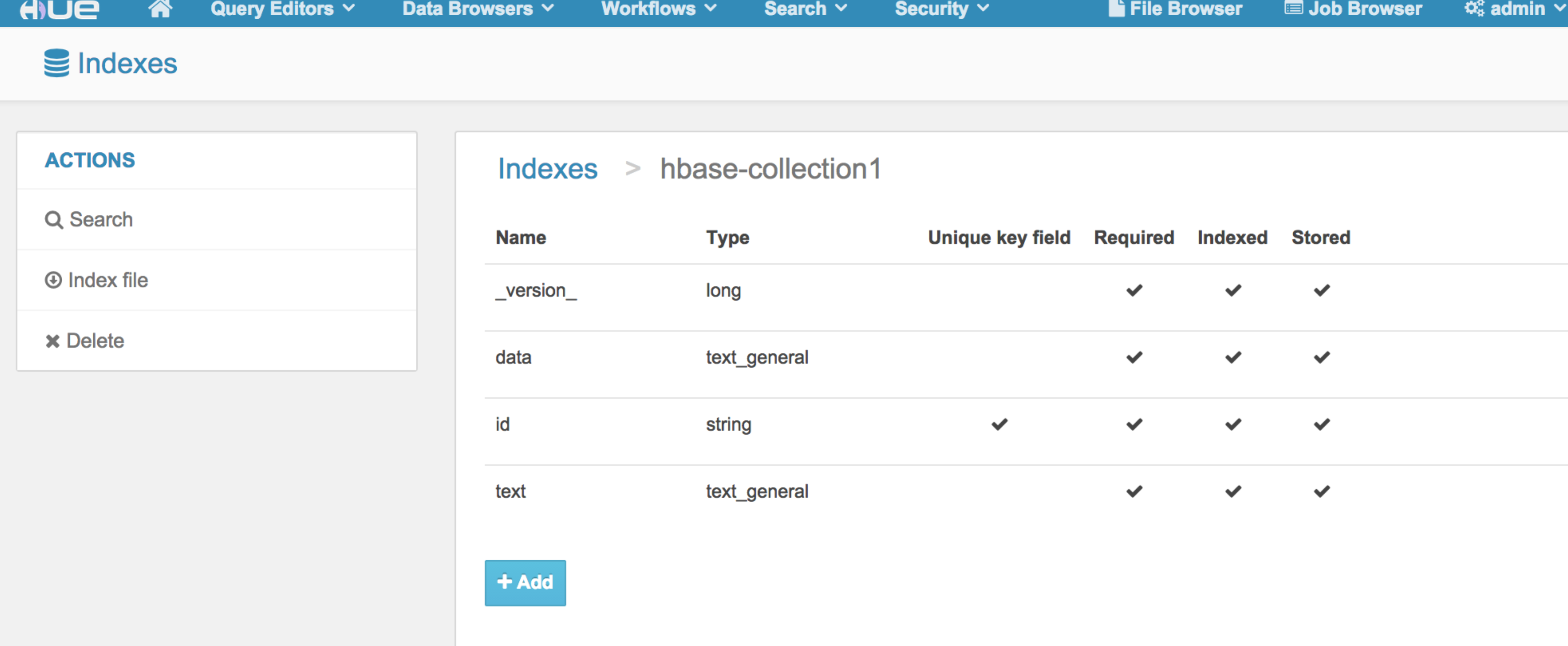
## Hue → Search 🡪 Indexes

Hue 🡪 Search 🡪 Indexes, you can see all the available indexes in Cloudera Search.



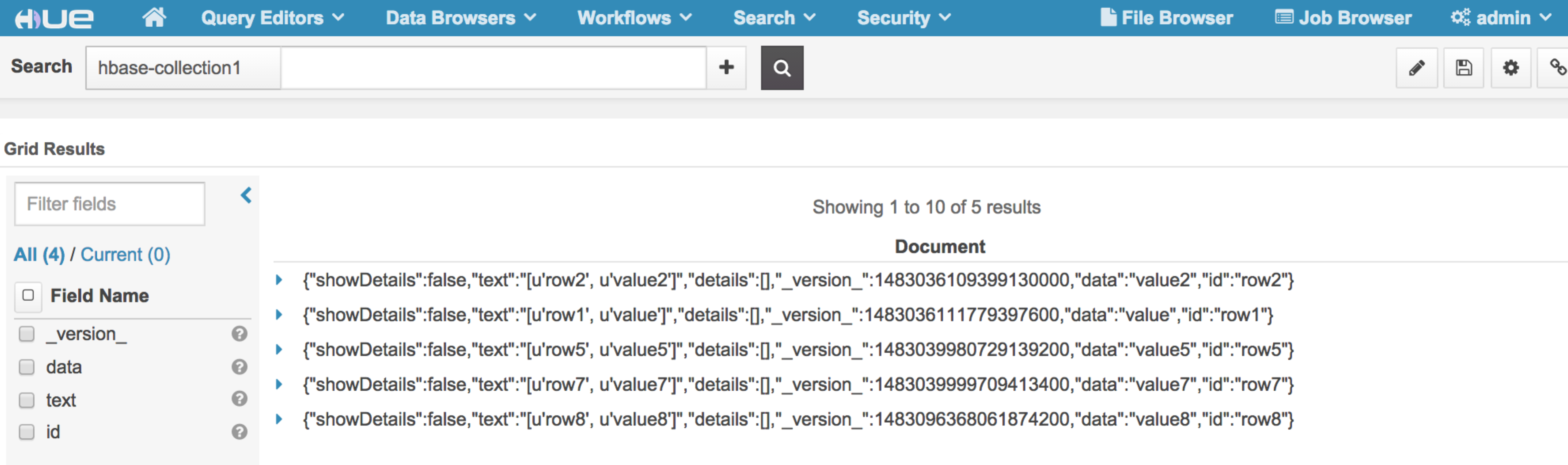
## Hue → Search 🡪 Indexes 🡪hbase-collection1

Hue 🡪 Search 🡪 Indexes🡪hbase-collection1, you can see all the available indexes definition for hbase-collection1.



## Hue → Search 🡪 Indexes 🡪 hbase-collection1🡪 Search

Hue 🡪 Search 🡪 Indexes🡪 hbase-collection1🡪 Search, you can see the Search Bar and, index Fields, and results for hbase-collection1.



## Search Keyword

Hue 🡪 Search 🡪 Indexes🡪 hbase-collection1🡪 Search, Search keyword in Search Bar to run full text search, for example: “value7”.

