Quick Start : How to Run Sensefy

Introduction

This document will guide you to build Sensefy and it's dependencies and configure Sensefy to crawl documents in Alfresco, and provide advance enterprise search features.

Sensefy Dependencies

- Apache ManifoldCF 2.3 : Framework for connecting to and crawling content and security policies from source repositories. ManifoldCF supports a large number of repositories. Full list of supported repositories can be found online at : http://manifoldcf.apache.org/en_US/release-documentation.html.
- Apache Solr 5.3 : Search server to index and query content. The Solr project site can be accessed here : http://lucene.apache.org/solr/
- Apache Stanbol 0.12 : Semantic engine for content enhancements used by Sensefy to provide semantic search features. The Stanbol project can be accessed here : http://stanbol.apache.org/

Following sections will guide you to build, configure and run above 3 dependencies of Sensefy.

Sensefy

Main component of the solution is the Sensefy Jar. First you can build and start Sensefy Jar as below.

Download Sensefy Source

```
git clone https: //github .com /zaizi/sensefy .git
cd sensefy
git checkout develop
```

Build Sensefy

Use the below command to run executable JAR with only Sensefy Application(Sensefy API, Sensefy UI and Sensefy Auth Server).

```
mvn clean install -Pbuild-frontend,war,build-jar -Dmaven.test.skip=true
```

Now you need to download and configure ManifoldCF, Solr and Stanbol.

ManifoldCF 2.3

Build ManifoldCF-2.3

```
git clone https: //github .com /apache/manifoldcf .git
cd manifoldcf/
git checkout release-2.3-branch
mvn clean install
```

If build fails due to test failure, Run

```
mvn clean install -DskipTests= true
```

In order to create ManifoldCF distribution, run

```
ant make -core-deps
ant make -deps
ant build
```

This will create **dist** directory (from now on referred as **${MANIFOLD_INSTALL_DIR}**)

Build Connectors

Stanbol Connector has a dependency on Stanbol Client (https://github.com/zaizi/apache-stanbol-client). Zaizi maven repository already contains this JAR. But If you are using public maven repositories, you first need to build Stanbol Client.

We have developed 2 connectors;

1. Stanbol Enhancer connector : a transformation connector that connects to Stanbol enhancer chains and returns documents with named entity recognition enhancements.
2. Stanbol SolrWrapper connector : a output connector that indexes the enhanced documents in a Solr instance. It maintains documents, entities identified in the documents and entity properties in 3 separate Solr indexes.

Build Stanbol Client

```
git clone https: //github .com /zaizi/apache-stanbol-client .git
cd apache-stanbol-client
git checkout jaxrs-1.0
mvn clean install -DskipTests= true
```

Build Sensefy Connectors

```
git clone https: //github .com /zaizi/sensefy-connectors .git
cd sensefy-connectors
mvn clean install -DskipTests= true
```

Configure Connectors with ManifoldCF

Add the built Connectors to **${MANIFOLD_INSTALL_DIR}/connector-lib**

Copy Connectors

```
cp transformation /mcf-stanbol-connector/target/mcf-stanbol-connector-2 .3-jar-
with-dependencies.jar ${MANIFOLD_INSTALL_DIR} /connector-lib/
cp output /mcf-solrwrapperconnector/target/mcf-solrwrapperprocessorconnector-
connector-2 .3.jar ${MANIFOLD_INSTALL_DIR} /connector-lib
```

Add following properties to **${MANIFOLD_INSTALL_DIR}/connectors.xml**

```
<transformationconnector name= "Stanbol enhancer" class=
"org.zaizi.manifoldcf.agents.transformation.stanbol.StanbolEnhancer" />
<outputconnector name= "Solr Wrapper" class=
"org.zaizi.manifoldcf.agents.output.solrwrapper.SolrWrapperConnector" />
```

Configure Alfresco AMP

Build AMP and Client

```
git clone https: //github .com /zaizi/alfresco-indexer .git
cd alfresco-indexer-webscripts
mvn clean install -DskipTests= true
cd .. /alfresco-indexer-client
mvn clean install -DskipTests= tru
```

Copy and Apply the AMP to Alfresco

```
cp alfresco-indexer-webscripts /target/alfresco-indexer-webscripts .amp
${ALFRESCO_INSTALL_DIR} /amps
sh bin /apply_amps .sh -force
```

Replace existing alfresco indexer client in **${MANIFOLD_INSTALL_DIR}/connector-lib** with new one.

```
rm ${MANIFOLD_INSTALL_DIR} /connector-lib/alfresco-indexer-client-0 .7.0.jar
cp alfresco-indexer-client /target/alfresco-indexer-client .jar
${MANIFOLD_INSTALL_DIR} /connector-lib/
```

Starting ManifoldCF

```
cd ${MANIFOLD_INSTALL_DIR} /example
java -jar start.jar
```

Solr 5.3

Download Solr

```
wget www.eu.apache.org /dist/lucene/solr/5 .3.1 /solr-5 .3.1.zip
unzip solr-5.3.1.zip
```

Solr 5.3 is used as the content indexer in Sensefy. You can run Solr either as a stand-alone server or as a cloud deployment.

Run Standalone Solr

We are running Solr in port 8983

```
cd  ${SOLR_DIR}/bin
./solr start -p 8983
```

Create Solr Cores

Copy Sensefy index configuration from the Sensefy project location to Solr directory. There are 3 indexes in Sensefy.

1. primaryIndex : The main core that indexes the documents
2. entity : Semantic entities such as people, organization and places are stored here
3. entityType : The attributes of different types of semantic entities used for semantic enhancements are here.

```
cp -r sensefy-runner /config/solr-default-conf/primaryIndex/
${SOLR_INSTALL_DIR} /server/solr/configsets/primaryIndex

cp -r sensefy-runner /config/solr-default-conf/entity/ ${SOLR_INSTALL_DIR}
/server/solr/configsets/entity

cp -r sensefy-runner /config/solr-default-conf/entityType/ ${SOLR_INSTALL_DIR}
/server/solr/configsets/entityType
```

Execute the core creation commands for the above 3 cores from the `${SOLR_INSTALL_DIR}` root directory

```
./bin/ solr create_core -c primaryIndex -d
./server/solr/configsets/primaryIndex/conf -p 8983

./bin/solr create_core -c entity -d ./server/solr/configsets/entity/conf -p
8983

./bin/solr create_core -c entityType -d
./server/solr/configsets/entityType/conf -p 8983
```

If you want to run Sensefy with Solr cloud please follow below steps.

Run Solr Cloud with Zookeeper

Download Zookeeper

```
wget www.eu.apache.org /dist/zookeeper/zookeeper-3 .4.6 /zookeeper-3 .4.6. tar
.gz
tar -zxvf  zookeeper-3.4.6. tar .gz
```

Configure Zookeeper

```
cd zookeeper-3.4.6
mkdir zkdata
mkdir -p  zkdata /1/
mkdir -p  zkdata /2/
mkdir -p  zkdata /3/

touch zkdata /1/myid
touch zkdata /2/myid
touch zkdata /3/myid
```

Note : myid files should contain corresponding directory ids in file. For example 1/myid should contain 1.

Please put corresponding ids in myid files.

Create Configs for zookeeper

```
cd zookeeper-3.4.6 /conf
touch zoo.cfg
touch zoo2.cfg
touch zoo3.cfg
```

Configuration for zoo.cfg

```
dataDir=${ZK_INSTALL_DIR} /zkdata/1
clientPort=2181
initLimit=5
syncLimit=2
server.1=localhost:2888:3888
server.2=localhost:2889:3889
server.3=localhost:2890:3890
```

Configuration for zoo2.cfg

```
dataDir=${ZK_INSTALL_DIR} /zkdata/2
clientPort=2182
initLimit=5
syncLimit=2
server.1=localhost:2888:3888
server.2=localhost:2889:3889
server.3=localhost:2890:3890
```

Configuration for zoo3.cfg

```
dataDir=${ZK_INSTALL_DIR} /zkdata/3
clientPort=2183
initLimit=5
syncLimit=2
server.1=localhost:2888:3888
server.2=localhost:2889:3889
server.3=localhost:2890:3890
```

Starting Zookeeper ensemble

Start Zookeeper

```
cd zookeeper-3.4.6
sh bin /zkServer .sh start zoo.cfg
sh bin /zkServer .sh start zoo2.cfg
sh bin /zkServer .sh start zoo3.cfg
```

Copy index configuration to Solr

```
cp -r sensefy-runner /config/solr-default-conf/primaryIndex/
${SOLR_INSTALL_DIR} /server/solr/configsets/primaryIndex

cp -r sensefy-runner /config/solr-default-conf/entity/ ${SOLR_INSTALL_DIR}
/server/solr/configsets/entity

cp -r sensefy-runner /config/solr-default-conf/entityType/ ${SOLR_INSTALL_DIR}
/server/solr/configsets/entityType
```

Upload Configuration to Zookeeper

In SolrCloud configurations are managed by Zookeeper. We can use zkcli utility in solr to upload the configurations.

```
cd solr-5.3.1/

. /server/scripts/cloud-scripts/zkcli .sh -zkhost
localhost:2181,localhost:2182,localhost:2183 -cmd putfile /solr .xml server
/solr/solr .xml

. /server/scripts/cloud-scripts/zkcli .sh -zkhost
localhost:2181,localhost:2182,localhost:2183 -cmd upconfig -confname
primaryIndex -confdir server /solr/configsets/primaryIndex/conf

. /server/scripts/cloud-scripts/zkcli .sh -zkhost
localhost:2181,localhost:2182,localhost:2183 -cmd upconfig -confname entity -
confdir server /solr/configsets/entity/conf

. /server/scripts/cloud-scripts/zkcli .sh -zkhost
localhost:2181,localhost:2182,localhost:2183 -cmd upconfig -confname
entityType -confdir server /solr/configsets/entityType/conf
```

Starting SolrCloud

```
mkdir -p server /cloud/node1
mkdir -p server /cloud/node2
mkdir -p server /cloud/node3
mkdir -p server /cloud/node4

. /bin/solr start -c -s server /cloud/node1/ -z
localhost:2181,localhost:2182,localhost:2183 -p 8984

. /bin/solr start -c -s server /cloud/node2/ -z
localhost:2181,localhost:2182,localhost:2183 -p 8985

. /bin/solr start -c -s server /cloud/node3/ -z
localhost:2181,localhost:2182,localhost:2183 -p 8986

. /bin/solr start -c -s server /cloud/node4/ -z
localhost:2181,localhost:2182,localhost:2183 -p 8987
```

Creating Collections

```
. /bin/solr create_collection -c primaryIndex -n primaryIndex -shards 2 -
replicationFactor 2 -p 8984
. /bin/solr create_collection -c entity -n entity -shards 2 -replicationFactor
2 -p 8984
. /bin/solr create_collection -c entityType -n entityType -shards 2 -
replicationFactor 2 -p 8984
```

Stanbol

Build Stanbol

```
git clone https: //github .com /apache/stanbol .git
cd stanbol
git checkout origin /release-0 .12
export MAVEN_OPTS= "-Xmx1024m -XX:MaxPermSize=256M"
mvn clean install -Dmaven. test .skip= true -DskipTests= true

java -jar launchers /full/target/org .apache.stanbol.launchers.full-0.12.1-
SNAPSHOT.jar -p 9090
```

Apache ManifoldCF Configuration

In this section, we will learn how to configure ManifoldCF with Alfresco as a content repository and as a user authority provider and a stand-alone Solr server as the output connection.

Configuring Apache ManifoldCF with PostgreSQL

Apache ManifoldCF includes Derby as a demo database to get started with. But in production environment you need to use more powerful databases. It is highly recommended to use PostgreSQL database with Apache ManifoldCF. You can follow below steps to configure PostgreSQL with ManifoldCF.

1) Install PostgreSQL (https://wiki.postgresql.org/wiki/Detailed_installation_guides)

2) Use following settings for PostgreSQL

- A default database encoding of UTF-8
- *postgresql.conf* settings as described in the table below

standard_conforming_strings	on
shared_buffers	1024MB
checkpoint_segments	300
maintenance*work*mem	2MB
tcpip_socket	true
max_connections	400
checkpoint_timeout	900
datestyle	ISO,European
autovacuum	off

- *pg_hba.conf* settings to allow password access for TCP/IP connections from ManifoldCF
- A maintenance strategy involving cronjob-style vacuuming, rather than PostgreSQL autovacuum

3) Change the ManifoldCF properties file to use PostgreSQL database. ***Properties.xml*** file can be found in ***manifold-artifact-1.8.1/example*** directory.

org.apache.manifoldcf.databaseimplementationclass	org.apache.manifoldcf.core.database.DBInterfacePostgreSQL
org.apache.manifoldcf.dbsuperusername	Name of your Postgre superuser
org.apache.manifoldcf.dbsuperuserpassword	Postgre superuser password

Once installed ManifoldCF will be running at **http://host-ip:8345/mcf-crawler-ui**

We have included example configurations to ease the configuration process. Following sections describe detailed process of providing configurations.

Follow the following steps to properly configure ManifoldCF.

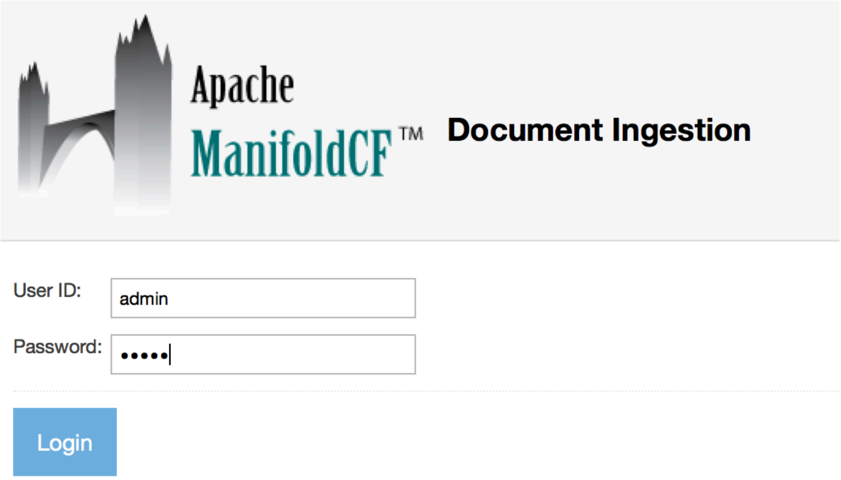
ManifoldCF with Alfresco repository connector and Solr Wrapper Connector

This section will guide you to index content from Alfresco repository to Solr with content enhanced by Stanbol to detect named entities in the content.

1. Login to ManifoldCF

Default User ID: admin

Default Password: admin



User ID: admin

Password: •••••

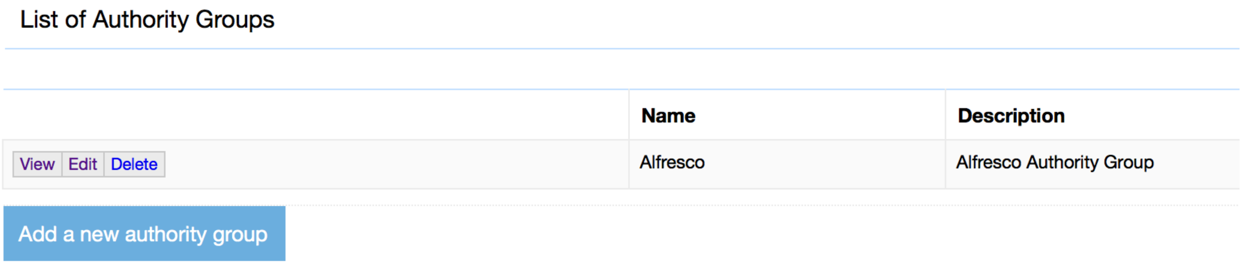
Once logged in you will be directed to following screen.

2. Configure Authority Connection for Alfresco

2.1 Create a Authority Group

2.1.1 Click on tab "List Authority Groups"

This tab will show the configured authority groups (If configured before)

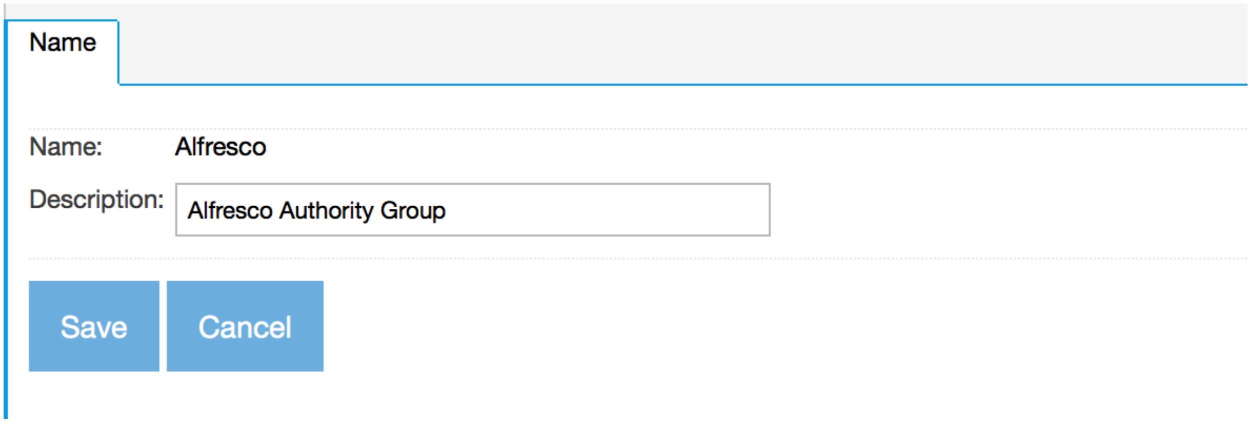


List of Authority Groups

	Name	Description
View Edit Delete	Alfresco	Alfresco Authority Group

2.1.2 In order to create new Authority Group, Click on "Add a new authority group"

2.1.3 Provide a name and a description and click on "Save"



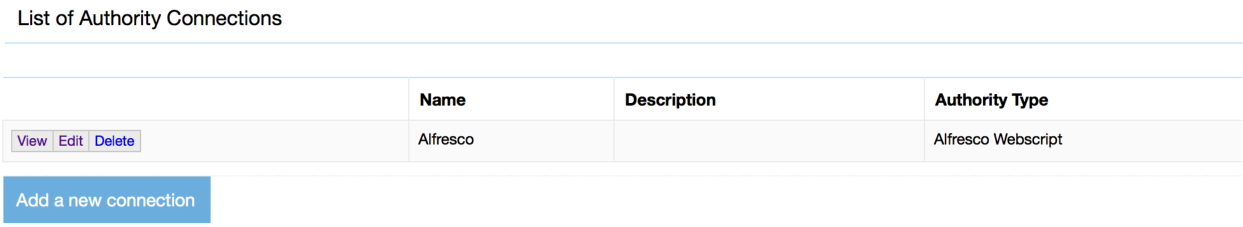
Name: Alfresco

Description: Alfresco Authority Group

2.2 Create a Authority Connection

2.2.1 Click on tab "List Authority Connection"

This tab will show the configured authority connections (If configured before)



List of Authority Connections

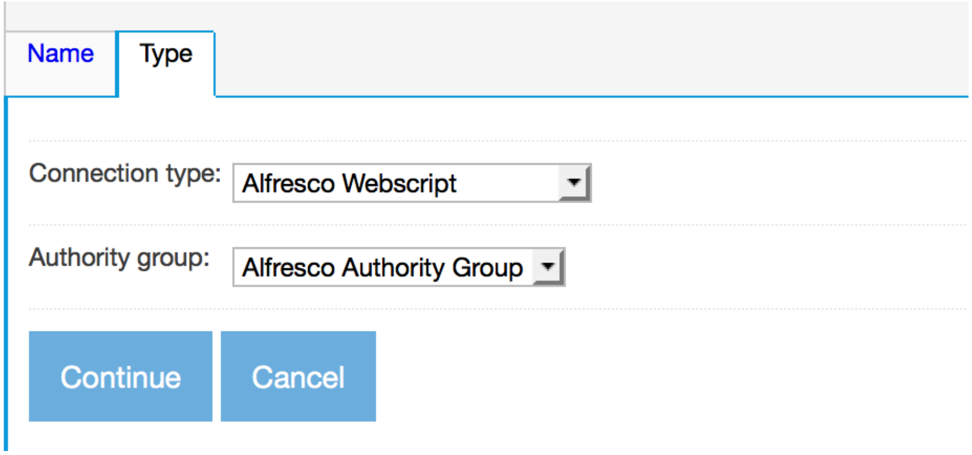
	Name	Description	Authority Type
View Edit Delete	Alfresco		Alfresco Webscript

2.2.2 In order to create new Authority Connection, Click on "Add a new connection"

2.2.3 Provide name and a description and click on "Type" tab

2.2.4 For "Connection type" select "Alfresco Webscript"

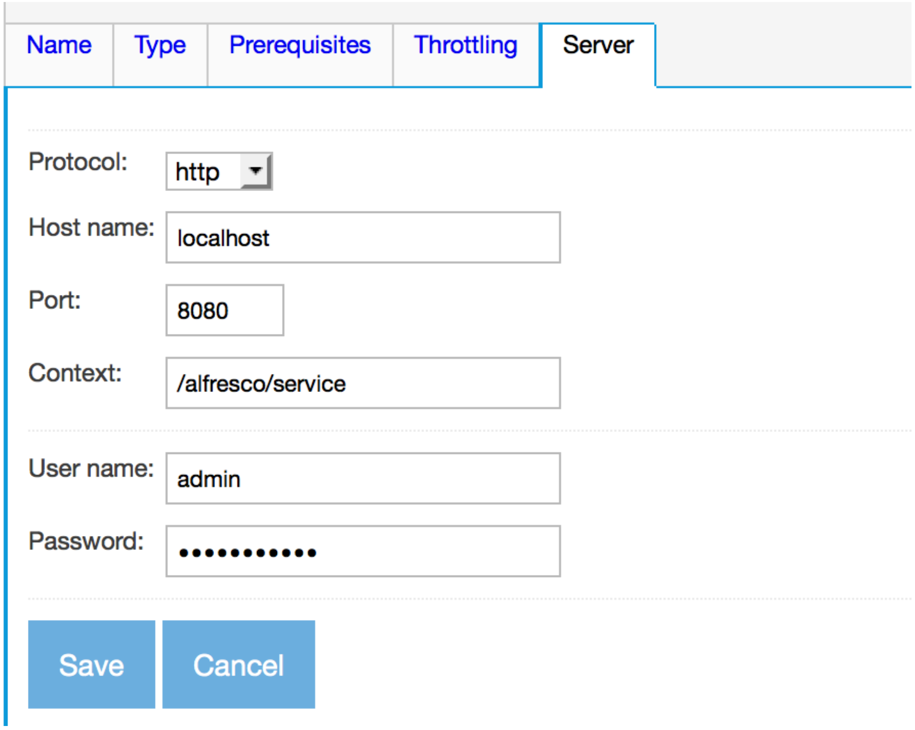
2.2.5 For "Authority group" select "Alfresco Authority Group" and click "Continue"



Connection type: Alfresco Webscript

Authority group: Alfresco Authority Group

2.2.6 Select tab "Server" and provide configuration for your Alfresco Server



Protocol: http

Host name: localhost

Port: 8080

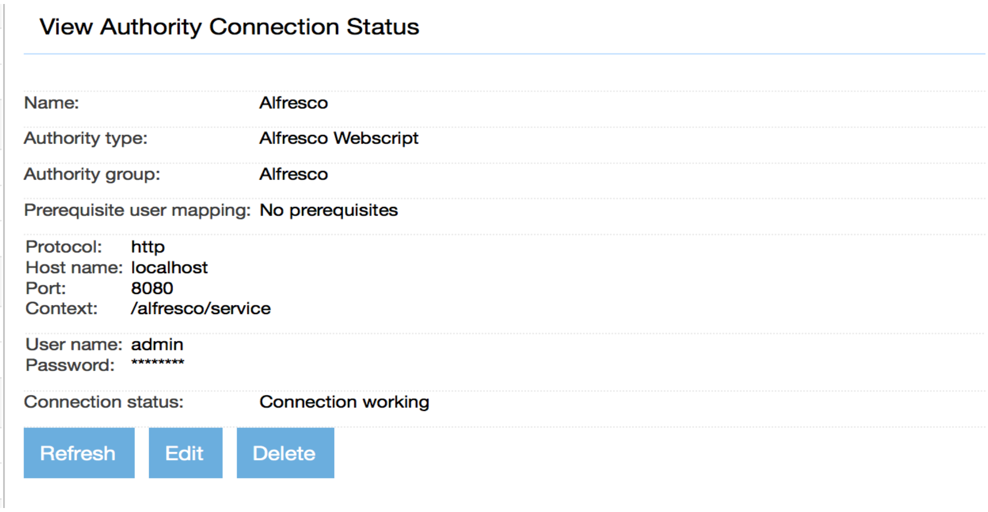
Context: /alfresco/service

User name: admin

Password: ●●●●●●●●●●●

2.2.7 Click on "Save"

If you properly configured authority connection, you will see the following status screen with connection status "Connection Working"



View Authority Connection Status

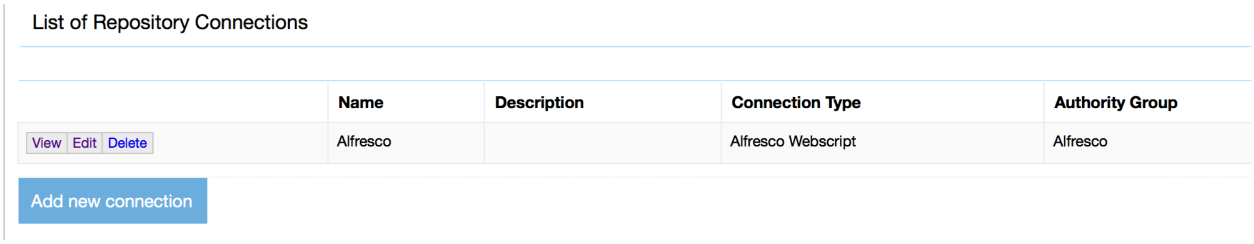
Name:	Alfresco
Authority type:	Alfresco Webscript
Authority group:	Alfresco
Prerequisite user mapping:	No prerequisites
Protocol: http Host name: localhost Port: 8080 Context: /alfresco/service	
User name: admin Password: ********	
Connection status:	Connection working

If something is wrong with Alfresco Server configuration connection status will give "Connection not working! Check configuration". If amp files are not properly installed connection will not work.

3. Configure Repository Connection for Alfresco

3.1 click on tab "List Repository Connections"

This tab will show the configured repository connections (If configured before)



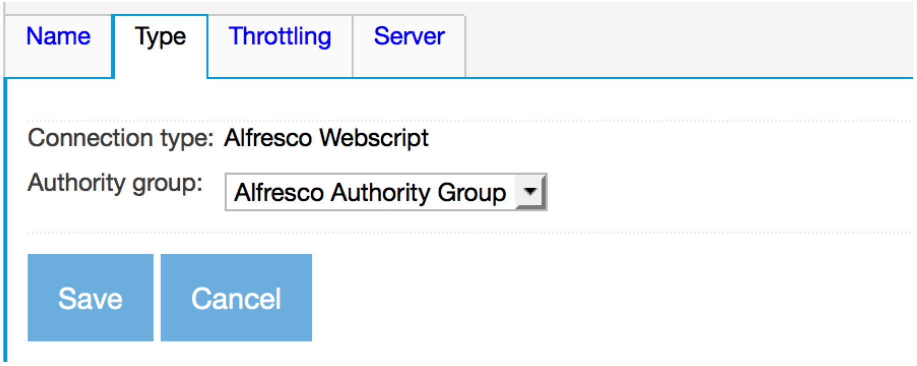
List of Repository Connections

	Name	Description	Connection Type	Authority Group
View Edit Delete	Alfresco		Alfresco Webscript	Alfresco

3.2 In order to create new Repository Connection, Click on "Add a new connection"

3.3 Provide a name and a description and click on tab "Type"

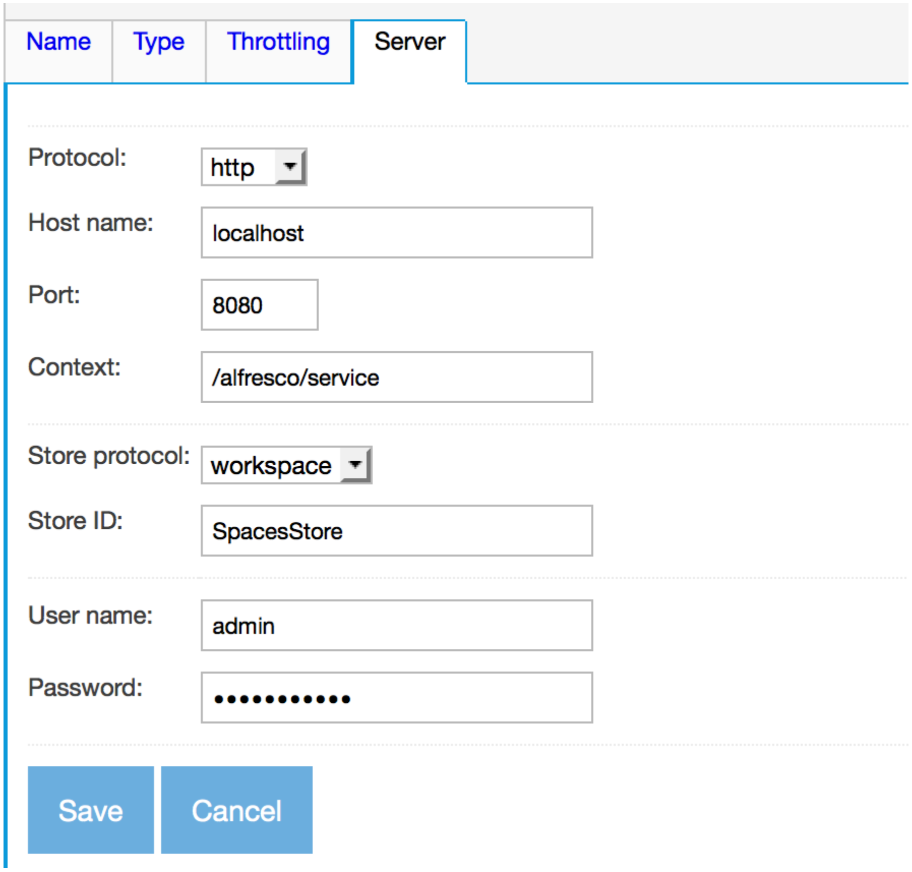
3.4 For Connection type select "Alfresco WebScript" and for Authority group select "Alfresco Authority Group" and Click on "Continue"



Connection type: Alfresco Webscript

Authority group: Alfresco Authority Group

3.5 Select tab "Server" and provide configuration for your Alfresco Server



Protocol: http

Host name: localhost

Port: 8080

Context: /alfresco/service

Store protocol: workspace

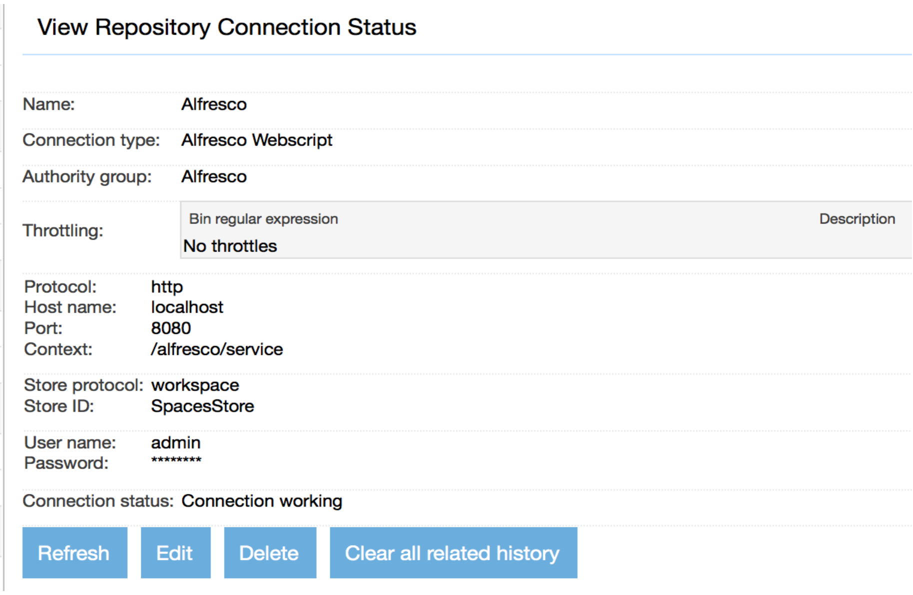
Store ID: SpacesStore

User name: admin

Password: ●●●●●●●●●●●

3.6 Click on "Save"

If you properly configured repository connection, you will see the following status screen with connection status "Connection Working"



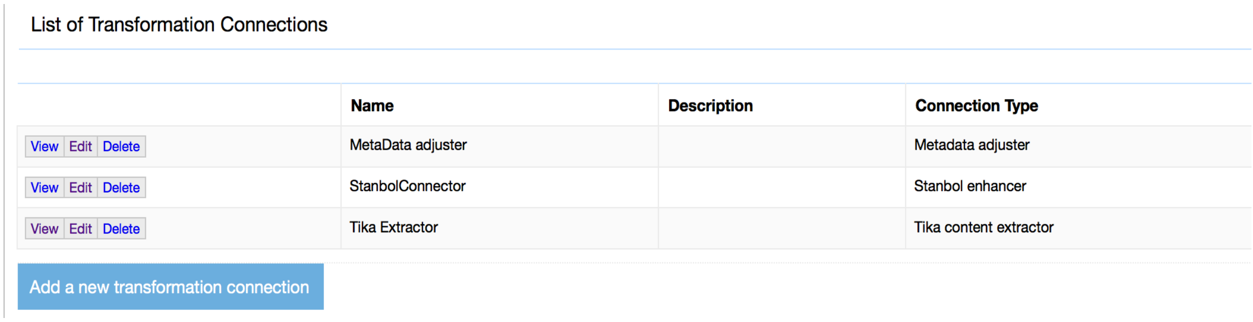
View Repository Connection Status

Name:	Alfresco
Connection type:	Alfresco Webscript
Authority group:	Alfresco
Throttling:	Bin regular expression / Description No throttles
Protocol: Host name: Port: Context:	http localhost 8080 /alfresco/service
Store protocol: Store ID:	workspace SpacesStore
User name: Password:	admin ********
Connection status:	Connection working

4. Configure Transformation Connections

Click on "List Transformation Connections".

This tab will show the configured transformation connections (If configured before)



List of Transformation Connections

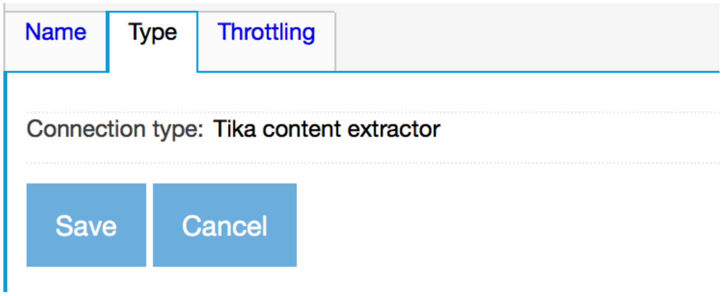
	Name	Description	Connection Type
View Edit Delete	MetaData adjuster		Metadata adjuster
View Edit Delete	StanbolConnector		Stanbol enhancer
View Edit Delete	Tika Extractor		Tika content extractor

4.1 Configure Tika Transformation Connection

4.1.1 Click on "Add a new transformation connection"

4.1.2 Provide a name and a description and click on "Type"

4.1.3 for Connection type select "Tika content extractor" and Continue



Connection type: Tika content extractor

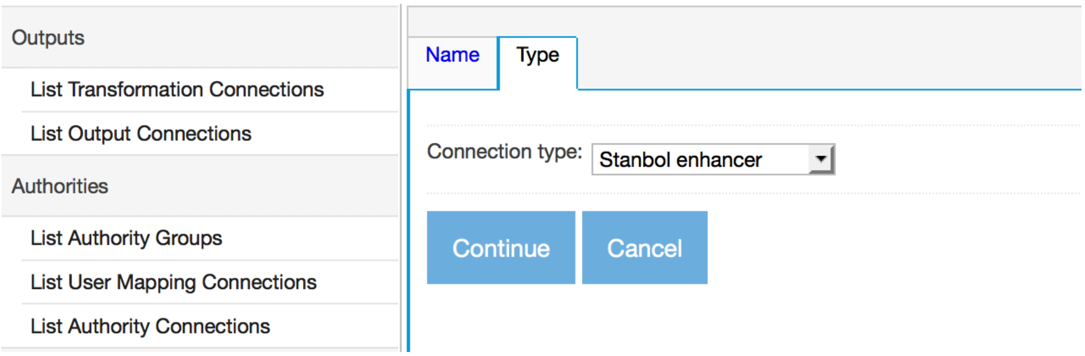
4.1.4 Click on "Save"

4.2 Configure Metadata Adjuster

Follow the same steps as for configuring Tika Transformation connection. In Type tab, for Connection Type select **"Metadata adjuster"**

4.3 Configure Stanbol Enhancer Transformation Connector

In type tab, for Connection Type select "Stanbol enhancer"


Outputs
List Transformation Connections
List Output Connections
Authorities
List Authority Groups
List User Mapping Connections
List Authority Connections
Name
Type
Continue
Cancel

Connection type: Stanbol enhancer

5. Configure SolrWrapper Output Connection

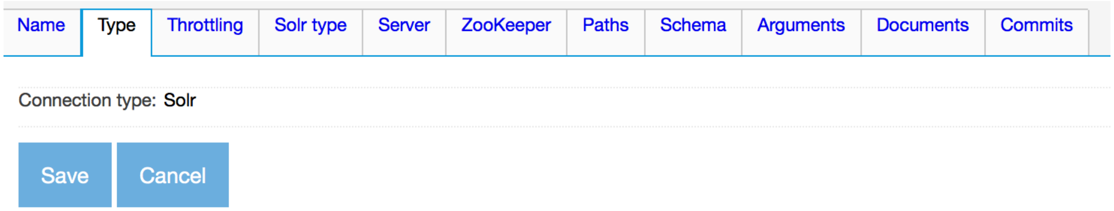
To configure SolrWrapper output connector, you first need to have 3 solr connectors configured to each Sensefy solr core (primaryIndex, entity and entityType). Therefore let's first create a Solr connector for primaryIndex.

5.2 Create a Solr Connector for primaryIndex

5.2.1. Click on "Add a new output connection"

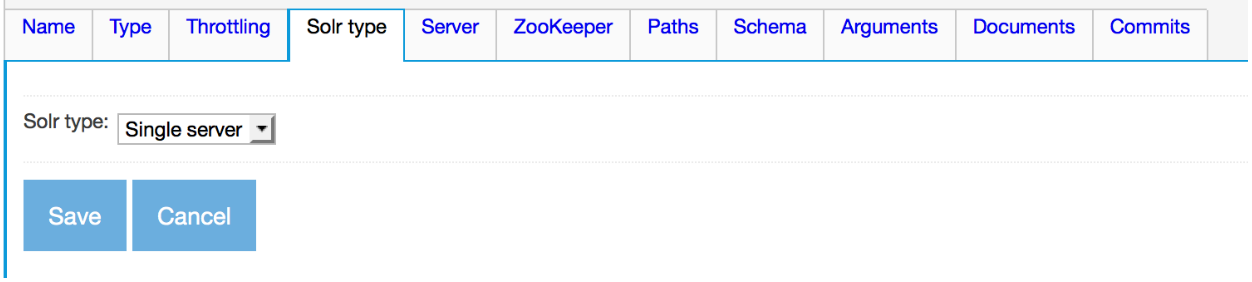
5.2.2 Provide name and a description and select Type tab

5.2.3 Select Solr as the Connection Type and Continue



Connection type: Solr

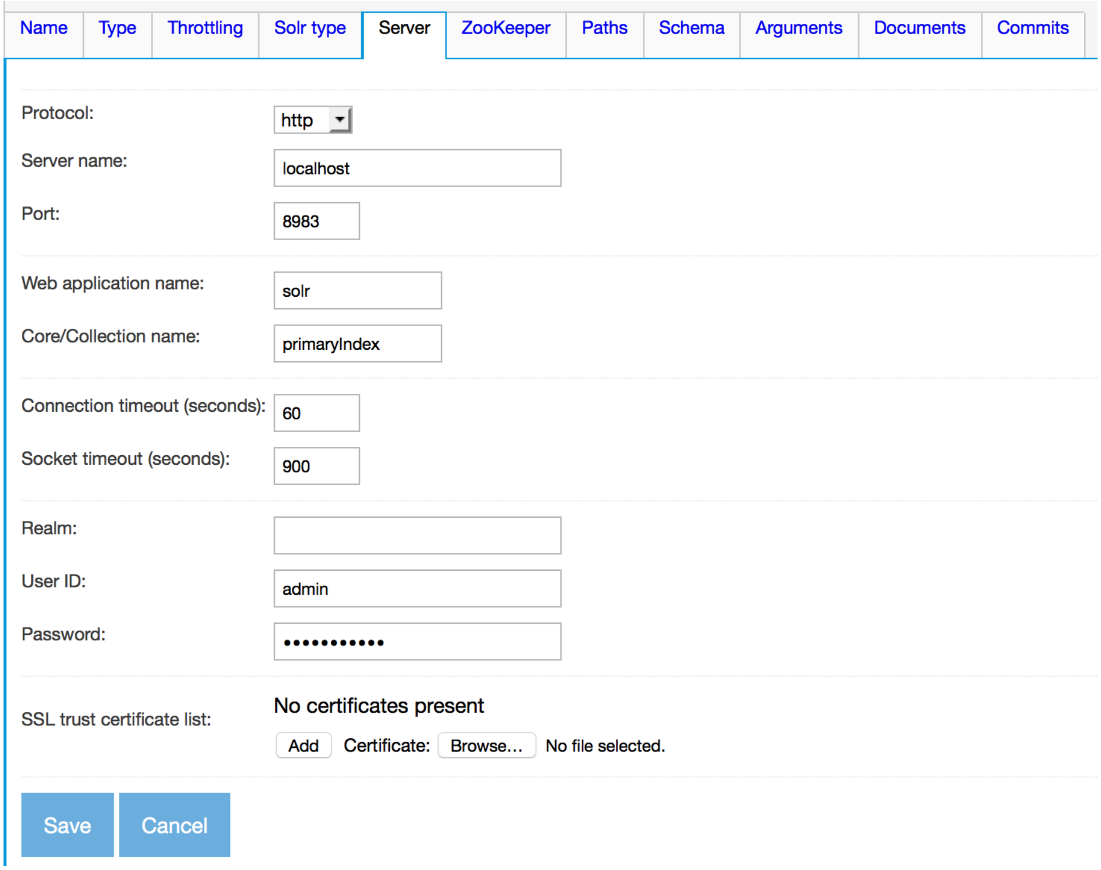
5.2.4 For Solr Type Select Single Server



Solr type: Single server

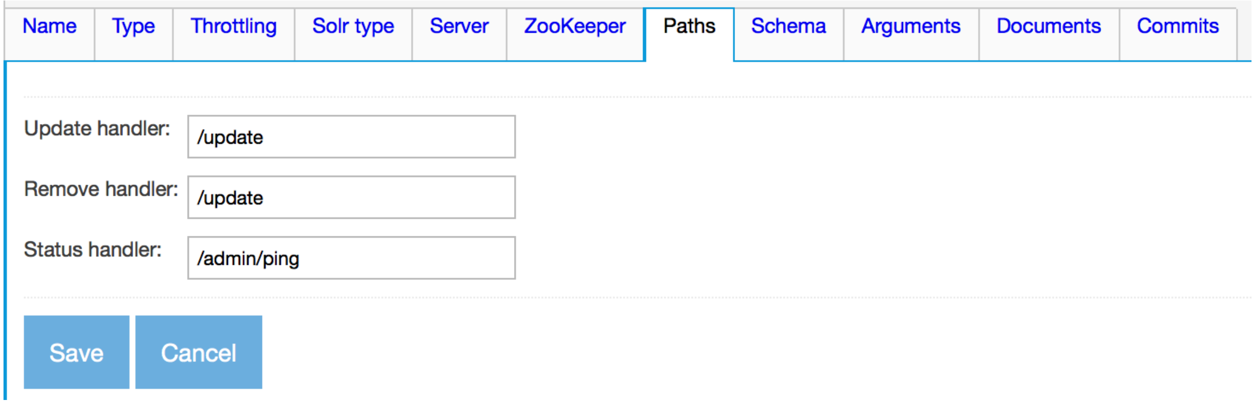
5.2.5 Provide Solr Server configuration parameters (This should point to Solr started in the installation step)

Important: For Core name provide PrimaryIndex



Protocol:	http
Server name:	localhost
Port:	8983
Web application name:	solr
Core/Collection name:	primaryIndex
Connection timeout (seconds):	60
Socket timeout (seconds):	900
Realm:	
User ID:	admin
Password:	•••••••••••
SSL trust certificate list:	No certificates present Certificate: No file selected.

5.2.6 Provide Path details as in the following diagram

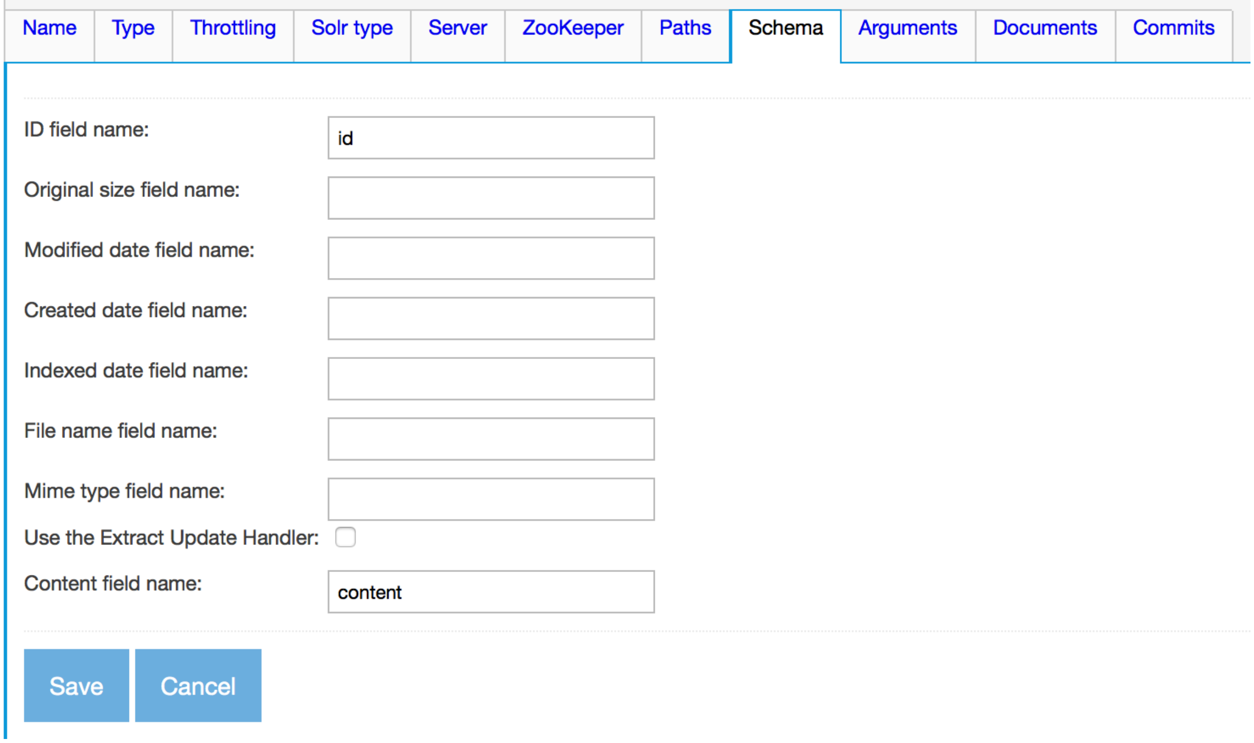


Update handler: /update

Remove handler: /update

Status handler: /admin/ping

5.2.7 Provide Schema details as in the following diagram



ID field name: id

Original size field name:

Modified date field name:

Created date field name:

Indexed date field name:

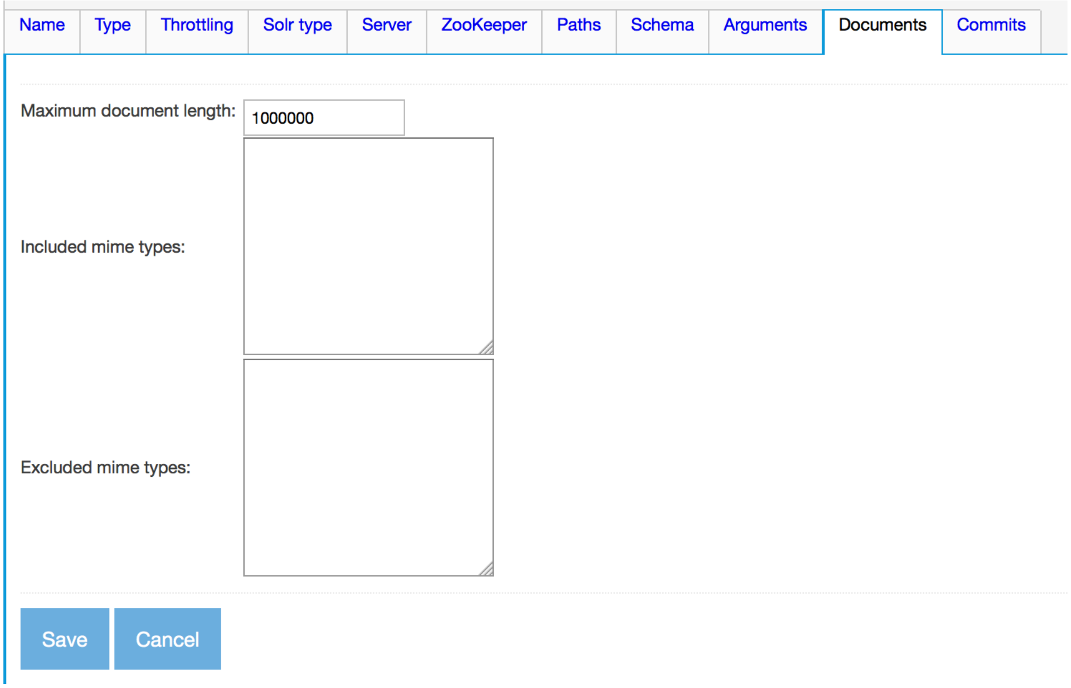
File name field name:

Mime type field name:

Use the Extract Update Handler: □

Content field name: content

5.2.8 Provide maximum document length for indexing



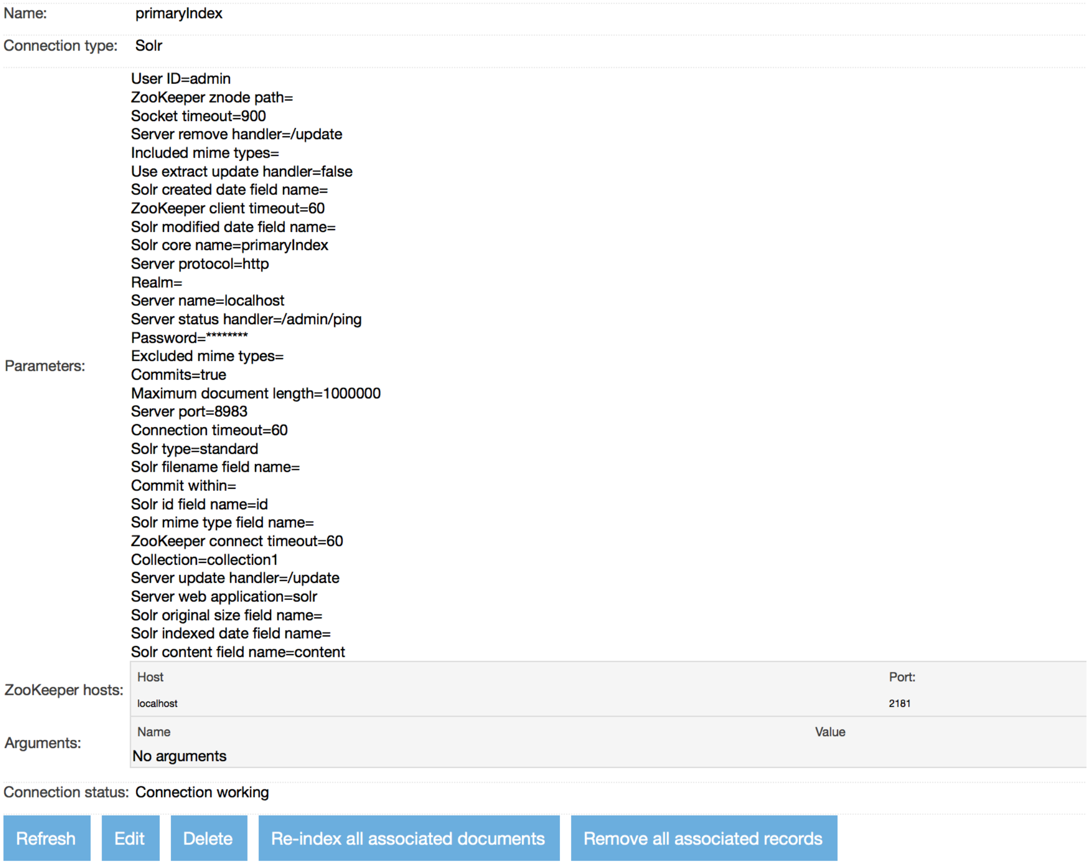
Maximum document length: 1000000

Included mime types:

Excluded mime types:

5.2.9 Click on "Save"

If you have properly configured Solr server, Following status page will appear.



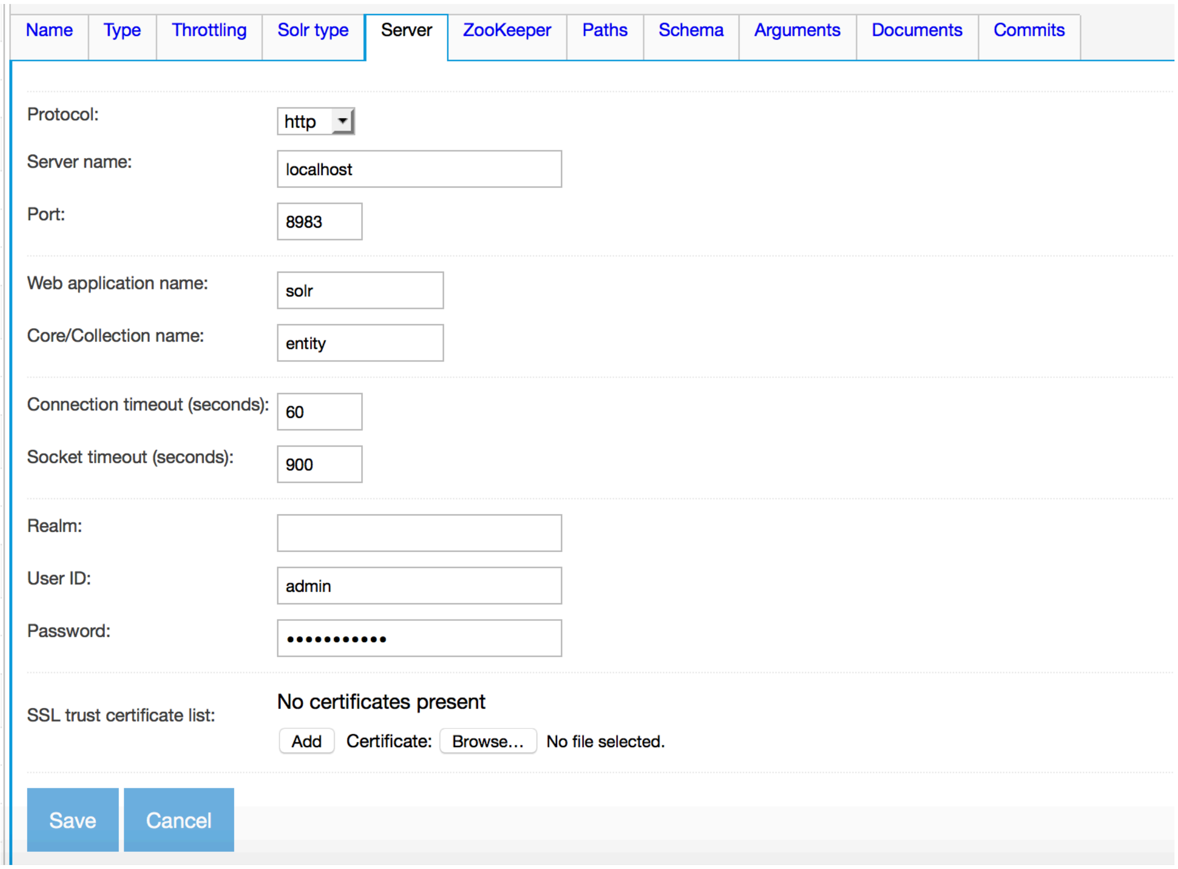
Name:	primaryIndex
Connection type:	Solr
Parameters:	User ID=admin ZooKeeper znode path= Socket timeout=900 Server remove handler=/update Included mime types= Use extract update handler=false Solr created date field name= ZooKeeper client timeout=60 Solr modified date field name= Solr core name=primaryIndex Server protocol=http Realm= Server name=localhost Server status handler=/admin/ping Password=******** Excluded mime types= Commits=true Maximum document length=1000000 Server port=8983 Connection timeout=60 Solr type=standard Solr filename field name= Commit within= Solr id field name=id Solr mime type field name= ZooKeeper connect timeout=60 Collection=collection1 Server update handler=/update Server web application=solr Solr original size field name= Solr indexed date field name= Solr content field name=content
ZooKeeper hosts:	Host: localhost Port: 2181
Arguments:	Name / Value No arguments
Connection status:	Connection working

Refresh
Edit
Delete
Re-index all associated documents
Remove all associated records

5.3. Create a Solr Connector for entity

Follow steps from 5.2.1 - 5.2.4

5.3.1. In the server configuration give Core name as "entity"



Protocol: http

Server name: localhost

Port: 8983

Web application name: solr

Core/Collection name: entity

Connection timeout (seconds): 60

Socket timeout (seconds): 900

Realm:

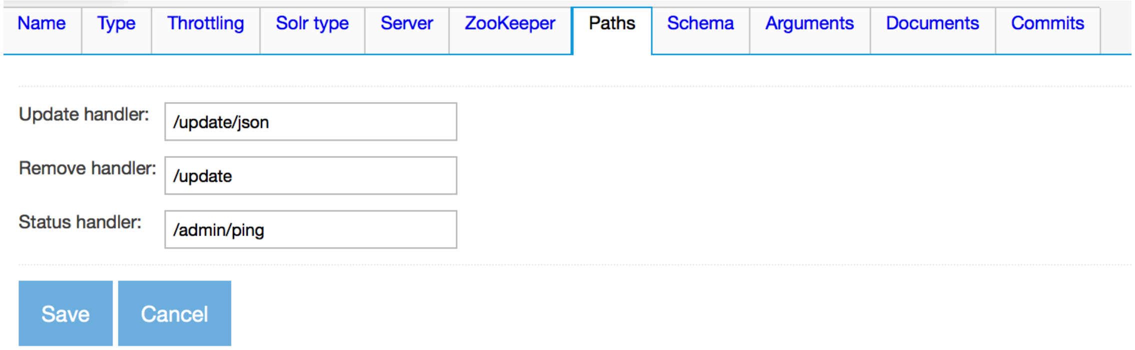
User ID: admin

Password: •••••••••••

SSL trust certificate list: No certificates present

Certificate: No file selected.

5.3.2 In Paths tab configure /update/json as the update handler

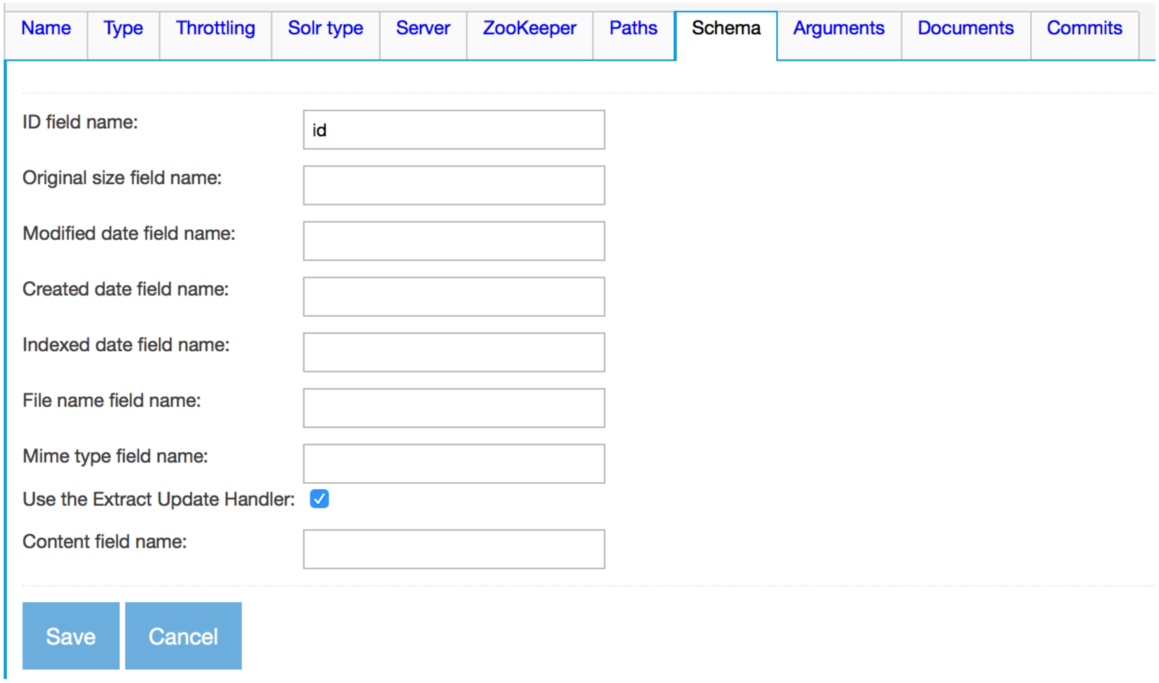


Update handler: /update/json

Remove handler: /update

Status handler: /admin/ping

5.3.3. In Schema tab enable Extract Update Handler



ID field name: id

Original size field name:

Modified date field name:

Created date field name:

Indexed date field name:

File name field name:

Mime type field name:

Use the Extract Update Handler: ☑

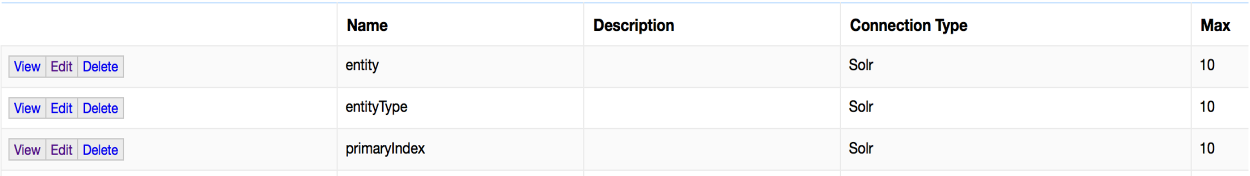
Content field name:

5.4 Create a Solr Connector for entityType

Follow the steps in 5.3 section and in the Schema tab give Core name as : entityType

5.5 Configure a SolrWrapper connector

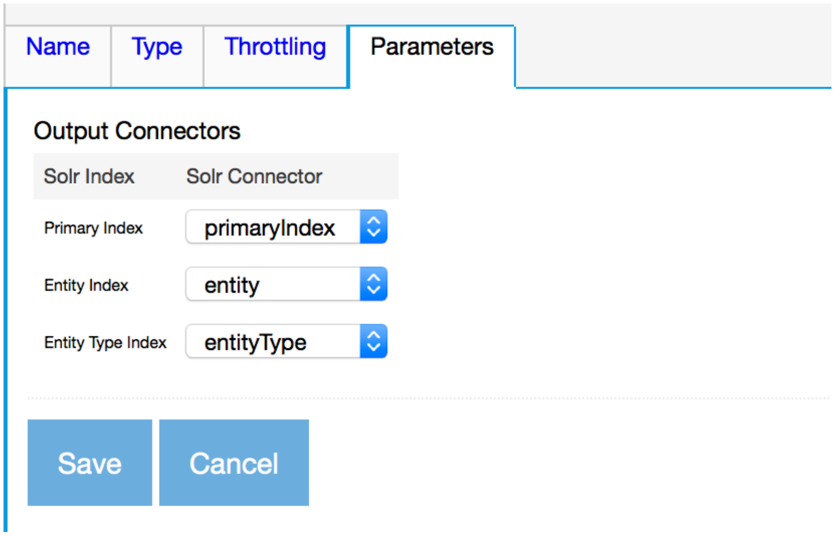
By now you have 3 solr connectors for each Solr core as below.



	Name	Description	Connection Type	Max
View Edit Delete	entity		Solr	10
View Edit Delete	entityType		Solr	10
View Edit Delete	primaryIndex		Solr	10

5.5.1. Add SolrWrapper type connector

5.5.2 In the parameters tab configure the Solr indexes for primary index, entity index, entity type index.



Output Connectors

Solr Index	Solr Connector
Primary Index	primaryIndex
Entity Index	entity
Entity Type Index	entityType

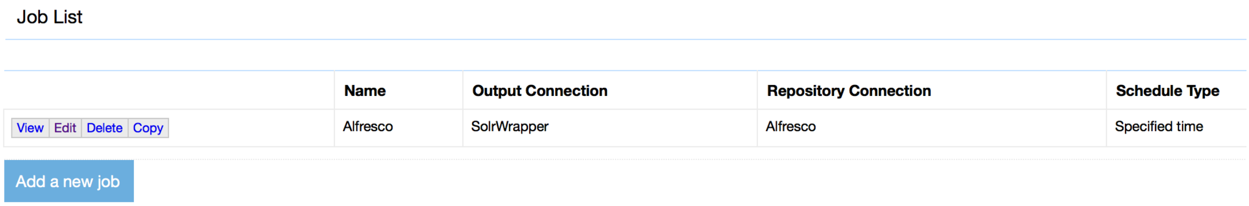
5.5.3 Click "Save" and save the SolrWrapper connector

6. Configure Jobs

Now let's configure a ManifoldCF job to crawl documents in Alfresco and index the enhanced content in Apache Solr index.

6.1 Click on "List all jobs"

This tab will show the configured jobs (If configured before)



Job List

	Name	Output Connection	Repository Connection	Schedule Type
View Edit Delete Copy	Alfresco	SolrWrapper	Alfresco	Specified time

6.2 Click "Add a new job" to create a new job & provide a name for the connection

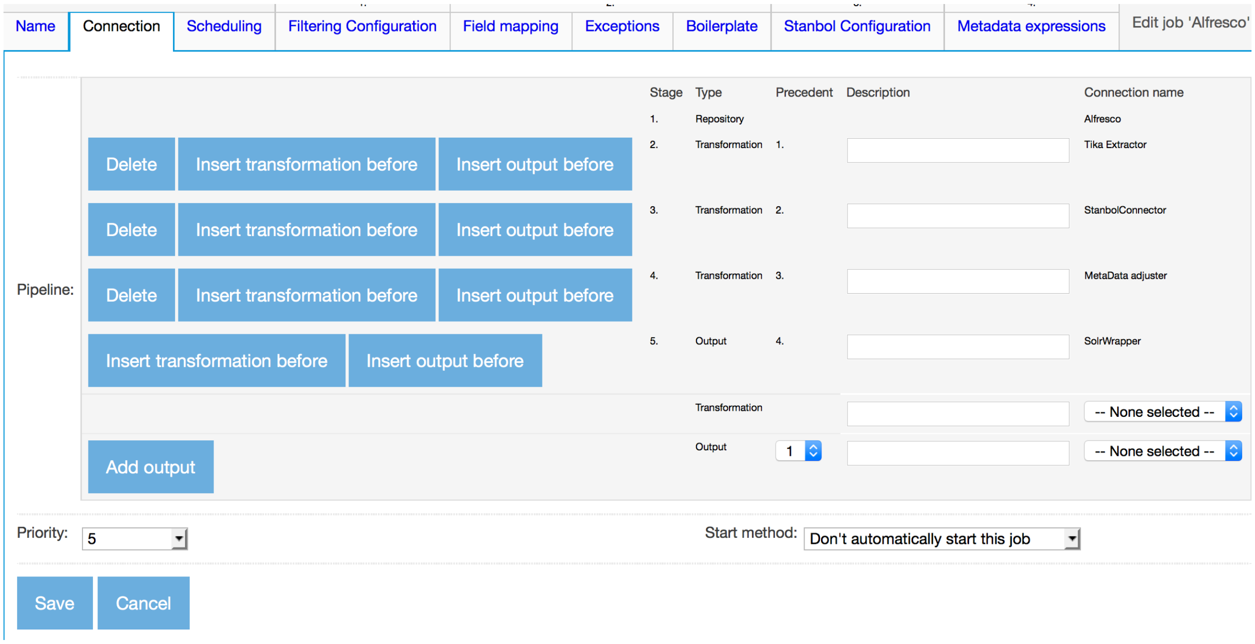
6.3 Configure Connections as shown in the following diagram

6.3.1 Add the required connectors to the connection.

Repository connector : Alfresco

Transformation connectors : TikaExtractor, StanbolConnector, MetaData adjuster

Output connector : SolrWrapper



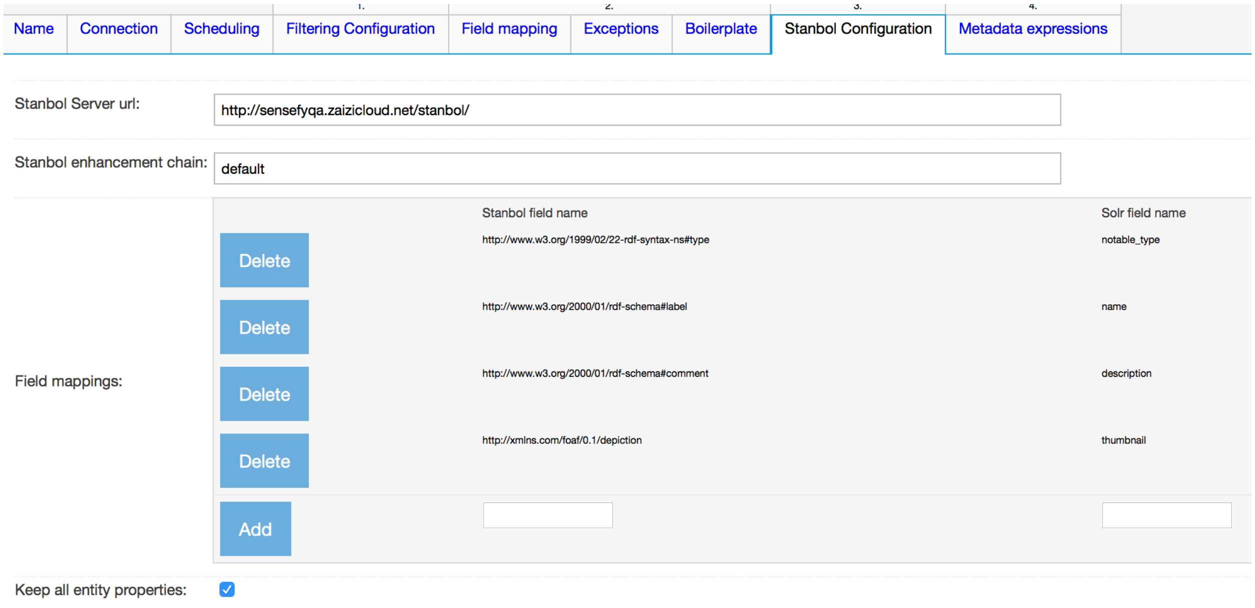
Pipeline:

Stage	Type	Precedent	Description	Connection name
1.	Repository			Alfresco
2.	Transformation	1.		Tika Extractor
3.	Transformation	2.		StanbolConnector
4.	Transformation	3.		MetaData adjuster
5.	Output	4.		SolrWrapper
	Transformation			-- None selected --
	Output	1		-- None selected --

Priority: 5

Start method: Don't automatically start this job

6.3.2 In the Stanbol Configuration tab define the Stanbol Server URL, enhancement chain and field mappings configurations.



Stanbol Server url: http://sensefyqa.zaizicloud.net/stanbol/

Stanbol enhancement chain: default

Field mappings:

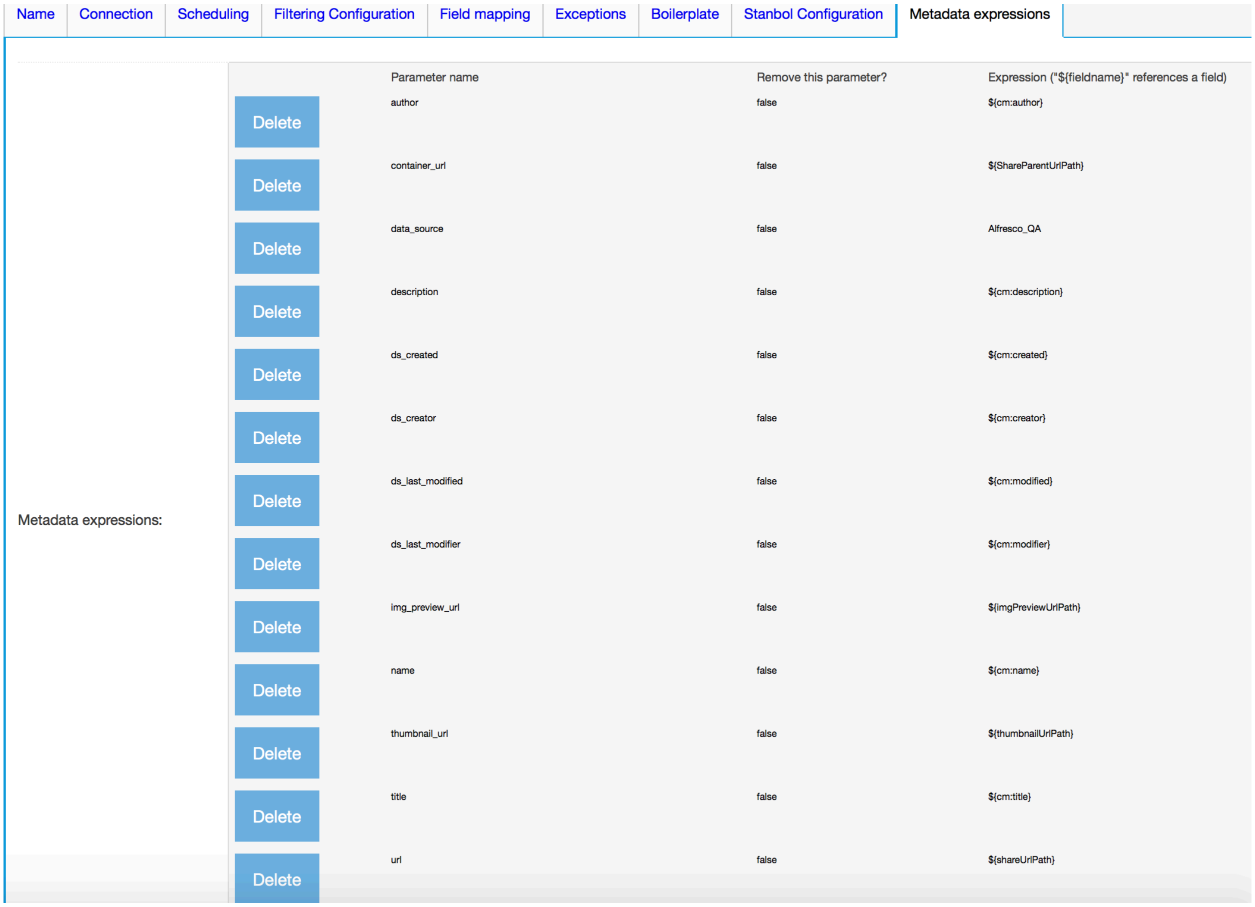
Stanbol field name	Solr field name
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	notable_type
http://www.w3.org/2000/01/rdf-schema#label	name
http://www.w3.org/2000/01/rdf-schema#comment	description
http://xmlns.com/foaf/0.1/depiction	thumbnail

Keep all entity properties: ☑

Recommended Field mappings for Stanbol for some common properties.

Stanbol Field Name	Solr Field Name
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	type
http://www.w3.org/2000/01/rdf-schema#label	name
http://www.w3.org/2000/01/rdf-schema#comment	description
http://xmlns.com/foaf/0.1/depiction	thumbnail

6.3.3 In Metadata expressions tab you have to map Alfresco metadata to output connection (In this case Solr) fields.



Metadata expressions:

	Parameter name	Remove this parameter?	Expression ("${fieldname}" references a field)
Delete	author	false	${cm:author}
Delete	container_url	false	${ShareParentUrlPath}
Delete	data_source	false	Alfresco_QA
Delete	description	false	${cm:description}
Delete	ds_created	false	${cm:created}
Delete	ds_creator	false	${cm:creator}
Delete	ds_last_modified	false	${cm:modified}
Delete	ds_last_modifier	false	${cm:modifier}
Delete	img_preview_url	false	${imgPreviewUrlPath}
Delete	name	false	${cm:name}
Delete	thumbnail_url	false	${thumbnailUrlPath}
Delete	title	false	${cm:title}
Delete	url	false	${shareUrlPath}

Recommended metadata expressions for Alfresco > SolrWrapper connection

Paramater Name	Remove this parameter?	Expression ${fieldname}
author	false	${cm:author}
container_url	false	${ShareParentUrlPath}
data_source	false	Alfresco_QA
description	false	${cm:description}
ds_created	false	${cm:created}
ds_creator	false	${cm:creator}
ds_last_modified	false	${cm:modified}
ds_last_modifier	false	${cm:modifier}
img_preview_url	false	${imgPreviewUrlPath}
name	false	${cm:name}
thumbnail_url	false	${thumbnailUrlPath}
title	false	${cm:title}
url	false	${shareUrlPath}

6.9 Click on "Save"

Run Sensefy

Before running Sensefy-2.0 you need to configure the Sensefy application properties. In order to override the default application properties built with sensefy, you need to add a config directory with Sensefy 2.0 jar with the application properties.

You can copy the 2 configuration files from ${sensefy git root}/sensefy-runner/config directory to a folder named as "config" in the Sensefy 2.0 jar class path.

Below are the application.properties and application.yml configuration files of Sensefy. You need to customize the solr endpoint, manifold.authority endpoint and alfresco endpoint with your URLs.

application.properties

```
#sensefy-api properties
spring.oauth2.resource.userInfoUri= http://localhost:9099/auth/user
spring.jmx.enabled=false

#solr properties
sensefy.search.solr.baseendpoint= http://localhost:8983/solr
sensefy.search.img.tempDir=./tempDir
sensefy.search.solr.cloud=false
sensefy.search.solr.zkEnsemble=localhost:2182,localhost:2181,localhost:2183
sensefy.token.ttl=18000000000
sensefy.token.secretkey=ad1ficultk3y

#sensefy-auth-conf properties
security.user.password=password123
logging.level.org.springframework.security=DEBUG
security.ignored= /css/**,/js/**,/favicon.ico,/webjars/**,/SensefyLogo.png
security.sessions=if-required

sensefy.authentication.alfresco.endpoint= http://localhost:8080/alfresco
sensefy.manifold.authority.endpoint= http://localhost:8345/mcf-authority-service
sensefy.search.endpoint= http://localhost:9099/search
sensefy.shares.domains=exampleDomain
endpoints.jmx.unique-names=true
spring.jmx.enabled=false
```

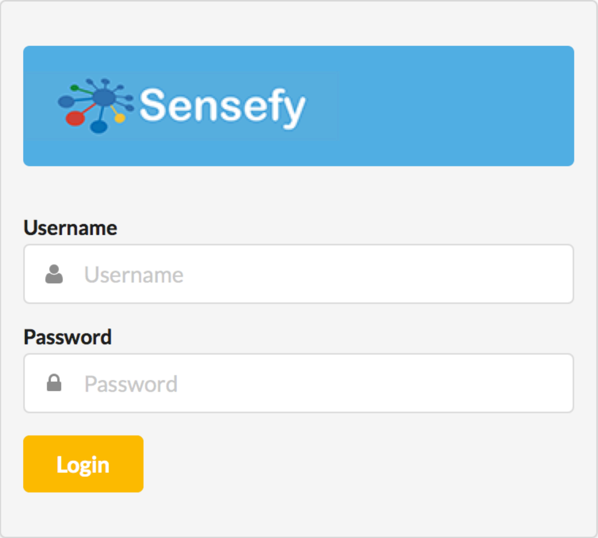
application.yml

```
debug:
security:
  user:
    password: password
  sessions: ALWAYS
  ignored: /css/**,/js/**,/favicon.ico,/webjars/**,/fonts/**
zuul:
  routes:
    resource:
      path: /service/api/**
      url:  http://localhost:9099/api
    user:
      path: /user/**
      url:  http://localhost:9099/auth/user
    oauth:
      path: /auth/**
      url:  http://localhost:9099/auth
    logout:
      path: /logout
      url:  http://localhost:9099/auth/logout
spring:
#  profiles: default
  oauth2:
    sso:
      logout-uri:  http://localhost:9099/auth/logout
      logout-redirect: true
      loginPath: /login
      home:
        secure: true
#        path: /**/*.html,/**/*.js,/**/*.css,/**/*.jpg,/**/*.png,/**/*.ico
    client:
      accessTokenUri:  http://localhost:9099/auth/oauth/token
      userAuthorizationUri:  http://localhost:9099t/auth/oauth/authorize
      clientId: sensefy
      clientSecret: sensefysecret
#      scope: openid
#      grant-type: authorization_code
#      clientAuthenticationScheme: form
#      pre-established-redirect-uri: http://localhost:8080
#      use-current-uri: true
    resource:
      userInfoUri: http://localhost:9099/auth/user
      preferTokenInfo: false
  jmx:
    enabled: false
endpoints:
  jmx:
    unique-names: true
logging:
  level:
    org.springframework.security: DEBUG
    org.springframework.web: DEBUG
```

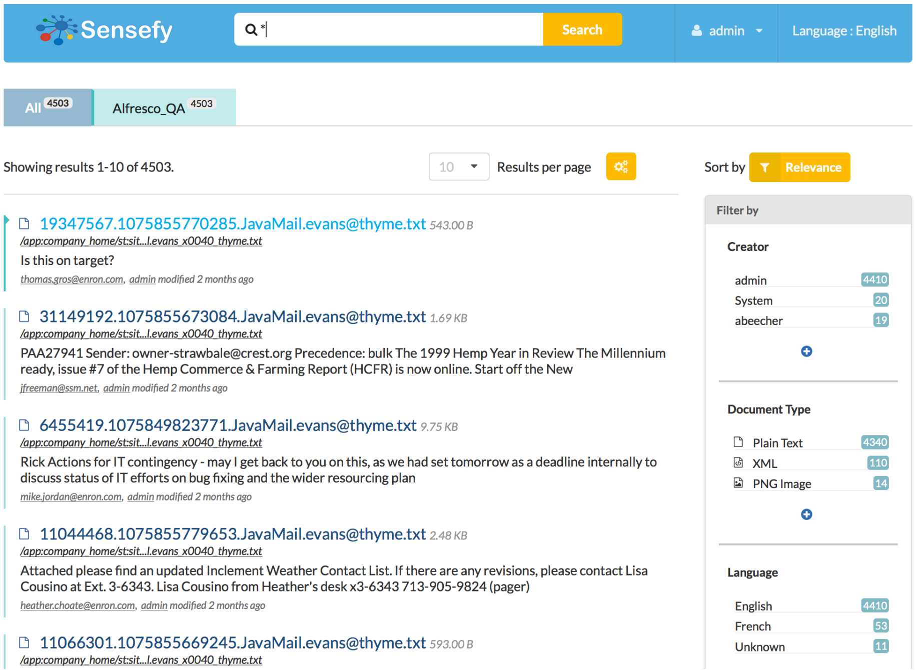
Run the Sensefy-2.0 jar as below

```
java -jar sensefy-2.0.2-SNAPSHOT.jar
```

You can now access Sensefy Search Login page using : http://localhost:9099



After login you will be directed to the Search, where you can perform content searches using advance search features given by Sensefy.



Showing results 1-10 of 4503.

19347567.1075855770285.JavaMail.evans@thyme.txt *543.00 B*
/app:company_home/st:sit...l.evans_x0040_thyme.txt
Is this on target?
thomas.gros@enron.com, admin modified 2 months ago

31149192.1075855673084.JavaMail.evans@thyme.txt *1.69 KB*
/app:company_home/st:sit...l.evans_x0040_thyme.txt
PAA27941 Sender: owner-strawbale@crest.org Precedence: bulk The 1999 Hemp Year in Review The Millennium ready, issue #7 of the Hemp Commerce & Farming Report (HCFR) is now online. Start off the New
jfreeman@ssm.net, admin modified 2 months ago

6455419.1075849823771.JavaMail.evans@thyme.txt *9.75 KB*
/app:company_home/st:sit...l.evans_x0040_thyme.txt
Rick Actions for IT contingency - may I get back to you on this, as we had set tomorrow as a deadline internally to discuss status of IT efforts on bug fixing and the wider resourcing plan
mike.jordan@enron.com, admin modified 2 months ago

11044468.1075855779653.JavaMail.evans@thyme.txt *2.48 KB*
/app:company_home/st:sit...l.evans_x0040_thyme.txt
Attached please find an updated Inclement Weather Contact List. If there are any revisions, please contact Lisa Cousino at Ext. 3-6343. Lisa Cousino from Heather's desk x3-6343 713-905-9824 (pager)
heather.choate@enron.com, admin modified 2 months ago

11066301.1075855669245.JavaMail.evans@thyme.txt *593.00 B*
/app:company_home/st:sit...l.evans_x0040_thyme.txt

Creator	
admin	4410
System	20
abeecher	19

Document Type	
Plain Text	4340
XML	110
PNG Image	14

Language	
English	4410
French	53
Unknown	11

You can access the ManifoldCF console at : **http://localhost:8345/mcf-crawler-ui**

You can access the Solr console at : **http://localhost:8983/solr**

You can access the Stanbol console at : **http://localhost:9090**