# Dataset Management: An Example

Jane is looking for data about products to study the trends in communication about the product and the life cycle of brands.

She needs data from onlie-markets and shop-systems, such as click statistics for product pages, web traces for users including order activity and social media data, such as tweets from Twitter to analyze the buzz and messages from Facebook.

First thing, she starts to collect twitter data using Flume.

* Add section on Flume data collection about the topic TTIP and CETA

## The tutorial about Twitter data collection

<http://www.tutorialspoint.com/apache_flume/fetching_twitter_data.htm>

Next, she starts to index the collected tweets. Technically this is simple using NRT indexing in FLUME. But it seems to be a good idea to also collect some more tweets about topics : … , … , … simply for tend comparison.

## Indexing AVRO data

The raw data has a schema: AVRO

SCHEMA

The collection has a schema: SOLR

SCHEMA

Both data sets should be discoverable by other people. Jane uses the Etosha exposure tool to expose the relevant metadata.

## Expose Metadata using Etosha CLI

Jane creates a dataset(descriptor) in the Metastore:

etosha create dataset **tweetsTTIP\_CETA**; // defines an entity (as page)

Now, she creates a flume-source (not Java code, but in the MD store) to track the configuration of the flume agent, which was used to collect the data:

etosha create datasource **flumeAgent01**; // defines an entity (as page)

A link is created between both like this:

etosha link **tweetsTTIP\_CETA** ‘**hasSource’** **flumeAgent01;** // links **S** and **O** via **P**

NOTE: ‘**hasSource’** has the role of an URI-shortcut if ‘ ’ is used. Plain text is interpreted directly as URI as in RDF syntax (Turtle triples).

Furthermore we add tags to the dataset:

Now we can add this data to the metastore:

etosha add tag to dataset **tweetsTTIP\_CETA to column col1 max=10000**

etosha add tag to dataset **tweetsTTIP\_CETA isAbout TTIP**

etosha add tag to dataset **tweetsTTIP\_CETA isAbout CETA**

One can see the pattern here: etosha add tag to dataset S P O

Currently, we use low level functions implemented in the etosha-core package to expose knowledge as facts represented by triples to a central location in HDFS. From here, the graph could be exported by external tools via Web-HDFS.

## How to Organize Context Information for Public Exposure?

The folder /etosha/cluster/local contains metadata about the cluster where this folder is stored in as N-Turtle files. All the files in this folder define the scope. Further enhancements are possible by time based partitions or by other subfolder structures which reflect domain specific best practices.

The initial implementation uses the default folder.

## The Turtle Syntax

As result we have this RDF file named etosha-core/rdf/demo1.rdf:

<flumeAgent01> <produces> "tweetsTTIP\_CETA" .

<flumeAgent01> <isOftype> "etosha:datasource" .

<tweetsTTIP\_CETA> <isAbout> "TTIP" .

<tweetsTTIP\_CETA> <isAbout> "CETA" .

<tweetsTTIP\_CETA> <hasColumn> "col1" .

<tweetsTTIP\_CETA> <isProducedby> "flumeAgent01" .

<tweetsTTIP\_CETA> <isOftype> "etosha:dataset" .

<col1> <isColumnIn> "tweetsTTIP\_CETA" .

<col1> <typeOfIdsUsed> "twitterUserAccounts" .

We load this RDF file into Fuseki in order to enable SPARQL queries on the exposed Metadata.

$ curl example for FUSEKI

Next, we create a DataFrame in Spark from the raw data in AVRO:

* Spark example …
* RDD describe function for columns

Some additional information like max, min, and number of distinct words are calculated.

## Next Steps:

<https://docs.google.com/document/d/1hPUm9SZAGEmK94854qEqM0dZIfkbYWKLodRyhINiOZk/edit#heading=h.t8q6ccruh16m>

[root@quickstart scripts]# ^C

[root@quickstart scripts]#

[root@quickstart scripts]# cat run.sh

#!/bin/bash

export FUSEKI\_HOME=/opt/fuseki/scripts/main/apache-jena-fuseki-2.3.0

export JAVA\_HOME=/opt/jdk1.8.0\_101

#

# Here we use a CLI parameter to define a location of the modelfile and the port.

#

CMD=$1

CORE\_MODEL\_FILE=/home/cloudera/.etosha/model.ttl

WEBSERVER\_PORT=9999

PART\_FOLDER=/home/cloudera/.etosha

#

# We need a time stamp

#

####### TS=$(time)

######################################

#

# Etosha Triple Collector Service

#

case $CMD in

(start)

clear

echo " FUSEKI\_HOME: $FUSEKI\_HOME"

echo " JAVA\_HOME: $JAVA\_HOME"

echo " PORT: $WEBSERVER\_PORT"

echo " CORE\_MODEL\_FILE: $CORE\_MODEL\_FILE"

echo " ETOSH-GRAPH-FOLDER: $PART\_FOLDER"

echo ">>> Starting the Fuseki-Server on port [$WEBSERVER\_PORT] (default: 3030)"

# clean

# ### BE CAREFULL ### rm -r $PART\_FOLDER -f

#

######## mv $PART\_FOLDER /tmp/$TS-ETOSHA-PARTITION-OLD

# load files to local from HDFS default location

######## hdfs dfs -get /etosha/snap/\* $PART\_FOLDER

exec $FUSEKI\_HOME/fuseki-server --file=$CORE\_MODEL\_FILE --update --port=$WEBSERVER\_PORT /ETCS &

sleep 5

FILES=$PART\_FOLDER/\*

for f in $FILES

do

echo "> LOAD ETOSH-GRAPH-SNAPHSOT: $f ..."

# take action on each file. $f store current file name

$FUSEKI\_HOME/bin/s-post http://localhost:$WEBSERVER\_PORT/ETCS/data default $f

done

;;

(list)

$FUSEKI\_HOME/bin/s-query --service http://localhost:$WEBSERVER\_PORT/ETCS/query 'SELECT \* {?s ?p ?o}'

;;

(reload)

;;

(\*)

echo "Don't understand [$CMD]"

;;

esac

root@quickstart scripts]# ./run.sh start

FUSEKI\_HOME: /opt/fuseki/scripts/main/apache-jena-fuseki-2.3.0

JAVA\_HOME: /opt/jdk1.8.0\_101

PORT: 9999

CORE\_MODEL\_FILE: /home/cloudera/.etosha/model.ttl

ETOSH-GRAPH-FOLDER: /home/cloudera/.etosha

>>> Starting the Fuseki-Server on port [9999] (default: 3030)

[2016-09-29 15:37:30] Server INFO Dataset: in-memory: load file: /home/cloudera/.etosha/model.ttl

[2016-09-29 15:37:31] Server INFO Fuseki 2.3.0 2015-07-25T17:11:28+0000

[2016-09-29 15:37:31] Config INFO FUSEKI\_HOME=/opt/fuseki/scripts/main/apache-jena-fuseki-2.3.0

[2016-09-29 15:37:31] Config INFO FUSEKI\_BASE=/var/run/cloudera-scm-agent/process/85-fuseki-FUSEKI\_SERVICE/scripts/run

[2016-09-29 15:37:31] Servlet INFO Initializing Shiro environment

[2016-09-29 15:37:31] Config INFO Shiro file: file:///var/run/cloudera-scm-agent/process/85-fuseki-FUSEKI\_SERVICE/scripts/run/shiro.ini

[2016-09-29 15:37:31] Config INFO Register: /ETCS

[2016-09-29 15:37:31] Server INFO Started 2016/09/29 15:37:31 PDT on port 9999

> LOAD ETOSH-GRAPH-SNAPHSOT: /home/cloudera/.etosha/model.ttl ...

[2016-09-29 15:37:34] Fuseki INFO [1] POST http://localhost:9999/ETCS/data?default

[2016-09-29 15:37:34] Fuseki INFO [1] POST /ETCS :: 'data' :: [text/turtle charset=UTF-8] ? default

[2016-09-29 15:37:34] Fuseki INFO [1] Body: Content-Length=12219, Content-Type=text/turtle, Charset=utf-8;charset=utf-8 => Turtle : Count=120 Triples=120 Quads=0

[2016-09-29 15:37:34] Fuseki INFO [1] 200 OK (60 ms)