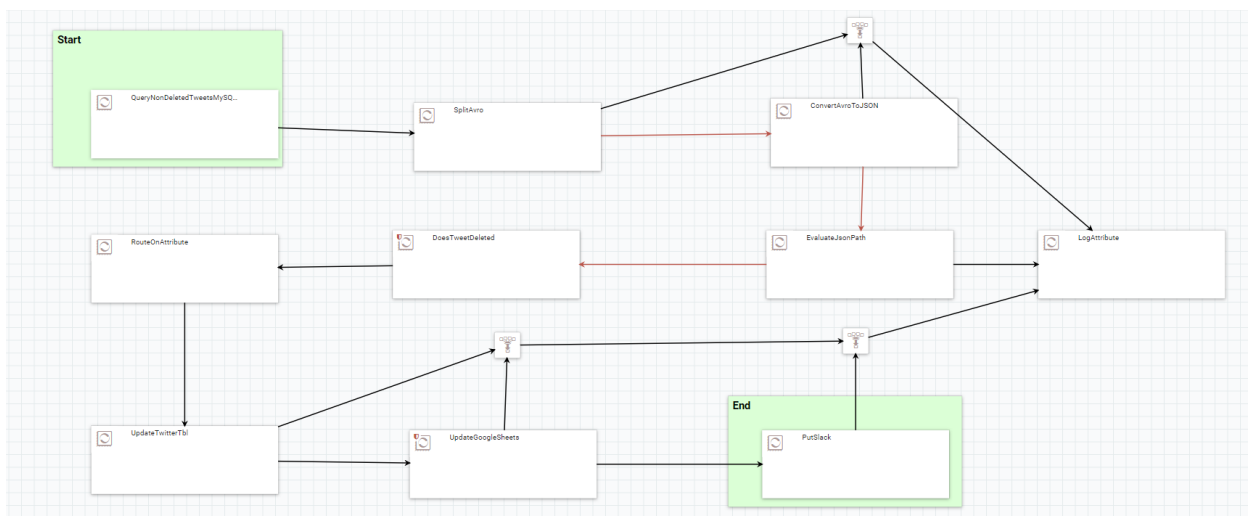# Mark Removed Tweets NiFi Process

Mark removed tweets process is taking undeleted tweets from MySQL table and check if the tweets were removed from Tweeter. If a tweet was removed, the tweet status both in MySQL and in Google Sheets will be changed to remove and a message will be sent to a slack channel/

Process Structure

Parameters

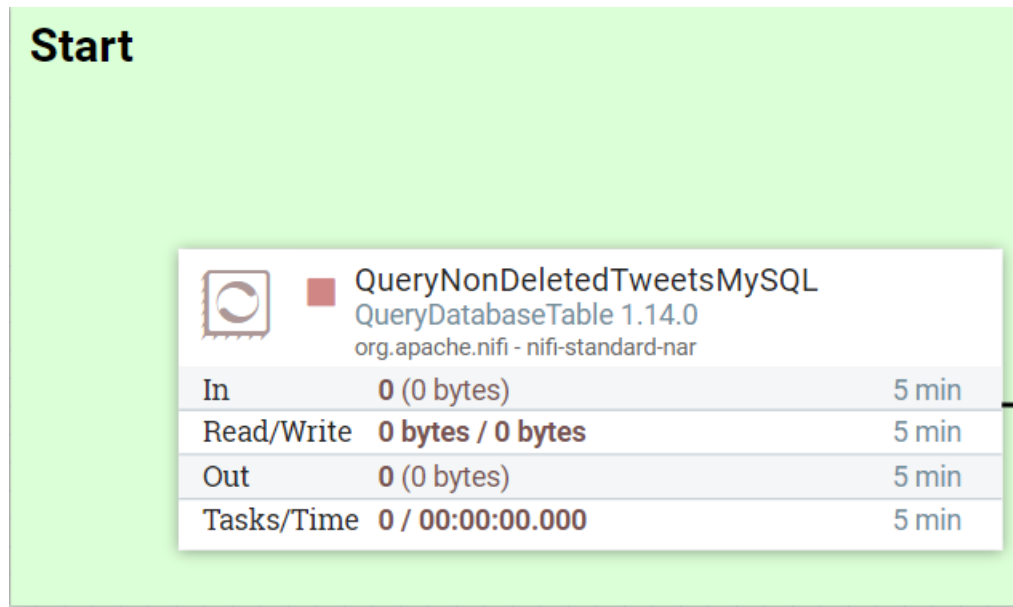| Name	Value	Sensitive Value
gsKeysFileName	/home/********/googleSheetkeys.json	
gsWorkbook	Reports	
gsWorksheet	Posts	
intervalDays	2	
mySQLHostname	****-****-*****	
mySQLPassword	**********	Y
mySQLPort	3306	
mySQLSchemaName	**********	
mySQLTwitterTable	twitterReportsTbl	
slChannel	adhoc_messages	
slWebhookURL	https://hooks.slack.com/services/*****/*****/********	Y
twAccessSecret	*********************************	
twAccessToken	*********************************	
twConsumerKey	*********************************	
twConsumerSecret	*********************************	

DBCPConnectionPooling

Property	Value
Database Connection URL	jdbc:mysql://#{mySQLHostname}:#{mySQLPort}/#{mySQLSchemaName}
Database Driver Class Name	com.mysql.jdbc.Driver
Database Driver Location	/opt/drivers/mysql-connector-java.jar
Database User	#{mySQLUsername}
Password	#{ mySQLPassword}

Mark Removed Tweets Process

markRemovedTweets

◎ 0 ⦸ 0 ▶ 3 ■ 8 ⚠ 0 ⚡ 0

Queued	**48,804** (21.95 MB)	
In	**0** (0 bytes) → **0**	5 min
Read/Write	**0 bytes / 0 bytes**	5 min
Out	**0** → **0** (0 bytes)	5 min

✔ 0 ✳ 0 ⬆ 0 ❶ 0 ? 0

QueryNonDeletedTweetsMySQL

Processor Name: QueryDatabaseTable

Description: Query from MySQL non deleted tweets.

Properties:

Property	Value
Database Connection Pooling Services	DBCPConnectionPool
Database Type	MySQL
Table Name	#{mySQLTwitterTable}
Columns to Return	tweetID
Additional Where Clause	logiRemoved = 'N' and reportingDate >= date_sub(now(), interval #{intervalDays} day)

Connection:

Relationship	From Process	To Process
Success	QueryNOonDeletedTweetsMySQL	SplitAvro

SplitAvro

Processor Name: SplitAvro

Properties:

Property	Value
Split Strategy	Record

Connection:

Relationship	From Process	To Process
Failure	SplitAvro	Funnel (LogAttribute)
Original	SplitAvro	
Split	SplitAvro	ConversionAvroToJson

ConvertAvroToJson

ConvertAvroToJSON
ConvertAvroToJSON 1.14.0
org.apache.nifi - nifi-avro-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Processor Name: ConvertAvroToJSON

Connection:

Relationship	From Process	To Process
Failure	ConversionAvroToJson	Funnel (LogAttribute)
Success	ConversionAvroToJson	EvaluateJsonPath

EvaluateJsonPath

Processor Name: EvaluateJsonPath

Description: NiFi variable tweetID initialize with mySQL tweetID

Properties:

Property	Value
Destination	flowfile-attribute
tweetID	$tweetID

Connection:

Relationship	From Process	To Process
Failure	EvaluateJsonPath	LogAttribute
matched	EvaluateJsonPath	DoesTweetDeleted
unmatched	EvaluateJsonPath	

DoesTweetDeleted

Processor Name: ExecuteStreamCommand

Description: Python script that checks in Twitter it the current tweetID is deleted

Note: GetTwitter NiFi process does not allow an incoming relationship, therefore ExecuteStreamCommand process was used.

Properties:

Property	Value
Command Argument Strategy	Command Argument Property
Commands Arguments	/home/naya/finalProject/NiFiCheckTweetStatus.py; #{twConsumerKey};#{twConsumerSecret};#{twAccessToken}; #{twAccessSecret};$.tweetID
Command Path	Python
Argument Delimiter	;
Output Destination Attribute	deleted

NiFiCheckTweetStatus.py

```
##
## NifiCheckTweetStatus.py
##
## The program is called from Nifi, DoesTweetDeleted (ExceuteStreamCommand Processor)
## The program checks if the tweet was deleted
##
## parameters:
##    1. Twitter's Customer key
##    2. Twitter's Customer secret
##    3. Twitter's Access Token
##    4. Twitter's Access  secret
##    5. Tweet ID
##
## Return values:
```

```
##      Y - Tweet was deleted
##      N - Tweet was not deleted

import sys
import tweepy as tw
#from tweepy import Stream
from tweepy import OAuthHandler


consumer_key = "{}".format(sys.argv[1])
consumer_secret = "{}".format(sys.argv[2])
access_token = "{}".format(sys.argv[3])
access_secret = "{}".format(sys.argv[4])
tweet_id = "{}".format(sys.argv[5])

# authenticate
auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)
api = tw.API(auth, wait_on_rate_limit=True)

# Check tweet status.
# api.statuses_lookup supported in old version of Python.
# In Python3 it should be changed to api.lookup_statuses
tweet_status = api.statuses_lookup([tweet_id])

if len(tweet_status) == 0:
    print ("Y")
else:
    print("N")
```

Connection:

Relationship	From Process	To Process
original	DoesTweetDeleted	RouteOnAttribute

RouteOnAttribute

Processor Name: RouteOnAttribute

Description: Continue in the flow only f tweet was deleted

Properties:

Property	Value
Routing Strategy	Route to Property Name
Yes	${deleted:contains('Y')}

Connection:

Relationship	From Process	To Process
Unmatched	RouteOnAttribute	
Yes	RouteOnAttribute	UpdateTwitterTbl

UpdateTwitterTbl

Processor Name: PutSQL

Description: Update Twitter Table with removed tweet

Properties:

Property	Value
JDBC Connection Pool	DBCPConnectionPool
SQL Statement	update #{mySQLTwitterTable} set logiRemoved = 'P', updatedVolDate = current_timestamp() where tweetID = ${tweetID}

Connection:

Relationship	From Process	To Process
failure	UpdateTwitterTbl	Funnel (LogAttribute)
retry	UpdateTwitterTbl	
success	UpdateTwitterTbl	UpdateGoogleSheets

UpdateGoogleSheets

Processor Name: ExecuteStreamCommand

Description: Python script to update Google Sheets with removed tweet

Properties:

Property	Value
Command Argument Strategy	Command Argument Property
Commands Arguments	/home/naya/finalProject/NiFiUpdateGsheets.py; #{gsKeysFileName};#{gsWorkbook};#{gsWorksheet};$.tweetID
Command Path	Python
Argument Delimiter	;

NiFiUpdateGsheets.py

```
## The program is called from Nifi, UpdateGoogleSheets (ExceuteStreamCommand Processor)
## The program update Google Sheets, when tweet is deleted
##
## parameters:
##    1. path and nam of Google Sheets Keys file
##    2. Google Sheets file name
##    3. Workbook name
##    4. Tweet ID
##


import sys
import pygsheets as gc

gsheets_keys = "{}".format(sys.argv[1])
gsheets_file_name = "{}".format(sys.argv[2])
workbook = "{}".format(sys.argv[3])
```

```
tweet_id = "{}".format(sys.argv[4])


# Connect to Google Sheets
gs = gc.authorize(service_account_file=gsheets_keys)
sh = gs.open(gsheets_file_name)
wks = sh.worksheet_by_title(workbook)

cells = wks.find(pattern=str(tweet_id), matchEntireCell=True, cols=[1, 1])

if (len(cells) != 0):
    row_no = cells[0].row
    print("Row Num:", row_no)

    if row_no > 0:
        cell_no = "C" + str(row_no)
        wks.update_value(cell_no, "Removed")
```
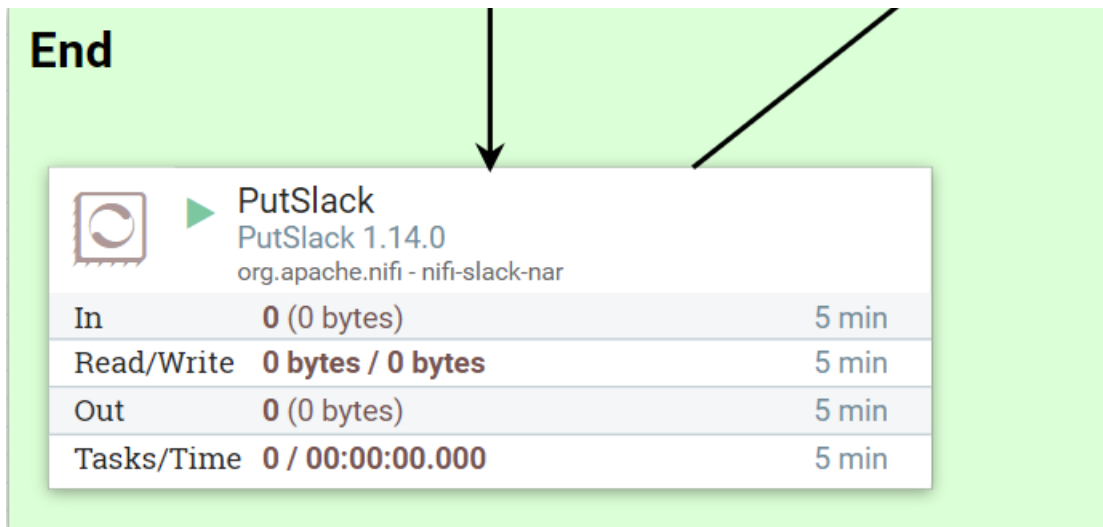
Connection:

Relationship	From Process	To Process
non zero status	UpdateGoogleSheets	Funnel (LogAttribute)
original	UpdateGoogleSheets	PutSlack
output stream	UpdateGoogleSheets	

PutSlack

Processor Name: PutSlack

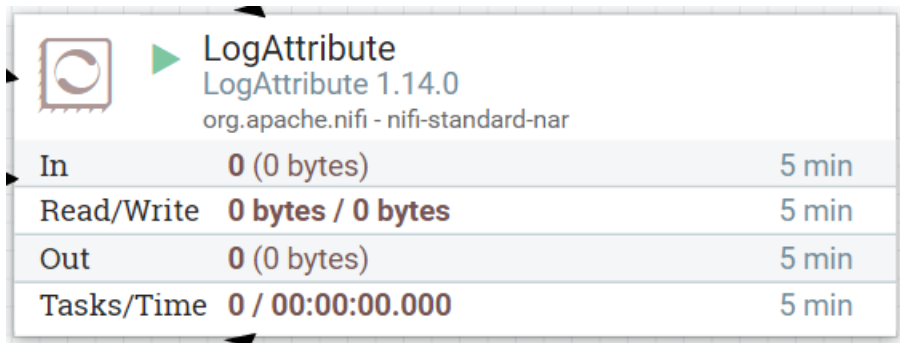**Description:** Send message to Slack channel about removed tweet

Properties:

Property	Value
Webhook URL	#{slWebhookURL}
Webhook Text	TweetID ${tweetID} was deleted
Channel	#{slChannel}

Connection:

Relationship	From Process	To Process
failure	PutSlack	Funnrl (LogAttribute)
success	PutSlack	

LogAttrinute

Processor Name: LogAttribute

Properties:

Property	Value
Log Leel	Info

Connection:

Relationship	From Process	To Process
success	LogAttribute	