Zeppelin

FINISHED

```
import org.apache.spark.streaming._
import org.apache.spark.streaming.twitter._
import org.apache.spark.storage.StorageLevel
import scala.io.Source
import scala.collection.mutable.HashMap
import java.io.File
import org.apache.log4j.Logger
import org.apache.log4j.Level
import sys.process.stringSeqToProcess
```

```
import org.apache.spark.streaming._
import org.apache.spark.streaming.twitter._
import org.apache.spark.storage.StorageLevel
import scala.io.Source
import scala.collection.mutable.HashMap
import java.io.File
import org.apache.log4j.Logger
import org.apache.log4j.Level
import sys.process.stringSeqToProcess
```

FINISHED

```
/** Configures the Oauth Credentials for accessing Twitter */
def configureTwitterCredentials(apiKey: String, apiSecret: String, accessToken: String, ac
  val configs = new HashMap[String, String] ++= Seq(
    "apiKey" -> apiKey, "apiSecret" -> apiSecret, "accessToken" -> accessToken, "accessTok
  println("Configuring Twitter OAuth")
  configs.foreach{ case(key, value) =>
    if (value.trim.isEmpty) {
      throw new Exception("Error setting authentication - value for " + key + " not set")
    }
    val fullKey = "twitter4j.oauth." + key.replace("api", "consumer")
    System.setProperty(fullKey, value.trim)
    println("\tProperty " + fullKey + " set as [" + value.trim + "]")
  }
  println()
}
```

```
configureTwitterCredentials: (apiKey: String, apiSecret: String, accessToken: String, acces
sTokenSecret: String)Unit
```

FINISHED

```
// Configure Twitter credentials
val apiKey = "EbUV1Oej9fLhmncj19IGvlXSf"
val apiSecret = "BbwOwLOgUyH3rPdIfE9WhCmu2wkbXqv2AukanQBD2pCtHob7M5"
val accessToken = "702953416281231362-MF8RPsZ7zcd1M9GhmnwkgTsncGKbpiD"
val accessTokenSecret = "W1aeYnbLrQa1a8xa9geXLPYDrsgMCQFL9Y7NsOUgk52il"
```

```
configureTwitterCredentials(apiKey, apiSecret, accessToken, accessTokenSecret)

import org.apache.spark.streaming.twitter._
val ssc = new StreamingContext(sc, Seconds(2))
val tweets = TwitterUtils.createStream(ssc, None)
val twt = tweets.window(Seconds(60))

case class Tweet(createdAt:Long, text:String)
twt.map(status=>
  Tweet(status.getCreatedAt().getTime()/1000, status.getText())
).foreachRDD(rdd=>
  // Below line works only in spark 1.3.0.
  // For spark 1.1.x and spark 1.2.x,
  // use rdd.registerTempTable("tweets") instead.
  rdd.toDF().registerTempTable("tweets")

)

twt.print

ssc.start()
```

```
StatusJSONImpl{createdAt=Thu Jan 26 19:24:56 EST 2017, id=824775161182314497, text='long le
g teen nude https://t.co/2gN55BgNpQ', source='<a href="http://twitter.com" rel="nofollow">T
witter Web Client</a>', isTruncated=false, inReplyToStatusId=-1, inReplyToUserId=-1, isFavo
rited=false, isRetweeted=false, favoriteCount=0, inReplyToScreenName='null', geoLocation=nu
ll, place=null, retweetCount=0, isPossiblySensitive=false, lang='nl', contributorsIDs=[], r
etweetedStatus=null, userMentionEntities=[], urlEntities=[URLEntityJSONImpl{url='https://t.
co/2gN55BgNpQ', expandedURL='https://goo.gl/vJvoj7', displayURL='goo.gl/vJvoj7'}], hashtagE
ntities=[], mediaEntities=[], symbolEntities=[], currentUserRetweetId=-1, user=UserJSONImpl
{id=258015287, name='Danielle Henderson', screenName='AsiaPasia', location='null', descript
ion='null', isContributorsEnabled=false, profileImageUrl='http://pbs.twimg.com/profile_imag
es/1264426767/DSCN7462fds_normal.jpg', profileImageUrlHttps='https://pbs.twimg.com/profile_
images/1264426767/DSCN7462fds_normal.jpg', isDefaultProfileImage=false, url='null', isProte
cted=false, followersCount=2, status=null, profileBackgroundColor='C0DEED', profileTextColo
r='333333', profileLinkColor='0084B4', profileSidebarFillColor='DDEEF6', profileSidebarBord
erColor='C0DEED', profileUseBackgroundImage=true, isDefaultProfile=false, showAllInlineMedi
a=false, friendsCount=4, createdAt=Sat Feb 26 14:39:00 EST 2011, favouritesCount=0, utcOffs
et=-1, timeZone='null', profileBackgroundImageUrl='http://pbs.twimg.com/profile_background_
images/306013819/thebackpack-close.jpg', profileBackgroundImageUrlHttps='https://pbs.twimg
```

READY

```
%sql select * from tweets where text like '%Donald Trump%' limit 60
```

FINISHED

createdAt	text
1,485,476,169	RT @OwenJones84: Unlike Donald Trump the majority of the British public are decent p
1,485,476,189	Absolutely unbelievable... "@GQMagazine: Donald Trump is discovering creative new w

```
%sql select createdAt, count(1) from tweets group by createdAt order by createdAt
```

FINISHED

createdAt
1,485,476,415
1,485,476,416
1,485,476,417
1,485,476,418
1,485,476,419
1,485,476,420
1,485,476,421
1,485,476,422
1,485,476,423

```
def sentiment(s:String) : String = {
    val positive = Array("like", "love", "good", "great", "happy", "cool", "the", "one", "
    val negative = Array("hate", "bad", "stupid", "is")

    var st = 0;

    val words = s.split(" ")
    positive.foreach(p =>
```

FINISHED

```
        words.foreach(w =>
            if(p==w) st = st+1
        )
    )

    negative.foreach(p=>
        words.foreach(w=>
            if(p==w) st = st-1
        )
    )
    if(st>0)
        "positivie"
    else if(st<0)
        "negative"
    else
        "neutral"
}
```

```
sentiment: (s: String)String
```

FINISHED

```
// Below line works only in spark 1.3.0.
// For spark 1.1.x and spark 1.2.x,
// use sqlc.registerFunction("sentiment", sentiment _) instead.
sqlc.udf.register("sentiment", sentiment _)
```

```
res27: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function
1>,StringType,Some(List(StringType)))
```

FINISHED

```
%sql select sentiment(text), count(1) from tweets where text like '%Donald Trump%' group by
```

UDF(text)	cour
neutral	3
positivie	1

READY