

Zeppelin

```
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 ▶ ⌕ 📖 ⚙️

```
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
```

```
/** Configures the OAuth Credentials for accessing Twitter */
def configureTwitterCredentials(apiKey: String, apiSecret: String, accessToken: String, accessTokenSecret: String)Unit = {
  val configs = new HashMap[String, String] ++= Seq(
    "apiKey" -> apiKey, "apiSecret" -> apiSecret, "accessToken" -> accessToken, "accessTokenSecret" -> accessTokenSecret
  )
  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()
}
```

FINISHED ▶ ⌕ 📖 ⚙️

```
configureTwitterCredentials: (apiKey: String, apiSecret: String, accessToken: String, accessTokenSecret: String)Unit
```

```
// Configure Twitter credentials
val apiKey = "ku7RowAjpJg57USGtHXut7miy"
val apiSecret = "32gpUALiPIdoMIu575tUzGvSF0aqHhggFK29vJG0LdmmU9DuMq"
val accessToken = "824757402067202048-nYZowB9PLlWLEyNtgpmL6U68ZBqR3vk"
```

FINISHED ▶ ⌕ 📖 ⚙️

```
val accessTokenSecret = "W4UsosNLQ04lF9h6GKuiEDKNk6nGibc98P1UbPSQWtTET"
configureTwitterCredentials(apiKey, apiSecret, accessToken, accessTokenSecret)
```

```
apiKey: String = ku7RowAjpJg57USGtHXut7miy
```

```
apiSecret: String = 32gpUALiPIdoMIu575tUzGvSF0aqHhqqFK29vJG0LdmmU9DuMq
```

```
accessToken: String = 824757402067202048-nYZowB9PLlWlEyNtgpmL6U68ZBqR3vk
```

```
accessTokenSecret: String = W4UsosNLQ04lF9h6GKuiEDKNk6nGibc98P1UbPSQWtTET
```

```
Configuring Twitter OAuth
```

```
Property twitter4j.oauth.consumerKey set as [ku7RowAjpJg57USGtHXut7miy]
```

```
Property twitter4j.oauth.accessToken set as [824757402067202048-nYZowB9PLlWlEyNtgpmL6U68ZBqR3vk]
```

```
Property twitter4j.oauth.consumerSecret set as [32gpUALiPIdoMIu575tUzGvSF0aqHhqqFK29vJG0LdmmU9DuMq]
```

```
Property twitter4j.oauth.accessTokenSecret set as [W4UsosNLQ04lF9h6GKuiEDKNk6nGibc98P1UbPSQWtTET]
```

FINISHED ▶ ⌘ 📖 ⚙

```
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()
```

```
import org.apache.spark.streaming.twitter._
ssc: org.apache.spark.streaming.StreamingContext = org.apache.spark.streaming.StreamingContext@4356f8b0
tweets: org.apache.spark.streaming.dstream.ReceiverInputDStream[twitter4j.Status] = org.apache.spark.streaming.twitter.TwitterInputDStream@706169a0
twt: org.apache.spark.streaming.dstream.DStream[twitter4j.Status] = org.apache.spark.streaming.dstream.WindowedDStream@11c29ce5
defined class Tweet
warning: there was one deprecation warning; re-run with -deprecation for details
```

```
%sql select * from tweets where text like '%Library%' limit 10
```

FINISHED ▶ ⌘ 📖 ⚙



```
%sql select * from tweets where text like '%girl%' limit 10
```

FINISHED ▶ 🔍 📖 ⚙️



createdAt	text
1,485,475,442	RT @GioMhirror: @xgirlInterrupted Ma l'hai visto non è un cesso [↑] TOP
1,485,475,443	RT @girlposts: everyone needs to watch this https://t.co/R91LIWY665
1,485,475,444	RT @WindowsXPmemes: That girl is https://t.co/ACClx0VoKm
1,485,475,448	RT @GawdTrill: If my girl aint hype about curving niggas like this i dont want he
1,485,475,449	RT @NapturalHair: I feel like every black girl can relate https://t.co/TEEx0yFefGo
1,485,475,450	RT @girlposts: I LOVE THIS 😭😭 https://t.co/yxf64F6rvi
1,485,475,451	RT @ThirtySecFights: Cash me ousside" girl gets cashed outside https://t.co/u
1,485,475,452	sexy school girls dad fuck ass pic https://t.co/WSY2xv3zFZ
1.485.475.459	RT @ThisDopeKid: when girls aet mad at vou https://t.co/0ZsYviHhaa

```
%sql select * from tweets where text like '%Donald Trump%' limit 20
```

FINISHED ▶ 🔍 📖 ⚙️



createdAt	text
1,485,475,585	RT @Salon: Demonstrators flood the streets of Philadelp
1,485,475,586	RT @Observador_CDMX: #Noticias HT Contra Donald Tr
1,485,475,587	RT @belindageoly: Donald Trump tient ses promesses de
1,485,475,592	Donald Trump, Şi Jinping ve küreselleşme
https://t.co/06WtHiMKRI	
Korkut Boratav yazdı	
https://t.co/zB1arBkGWN	

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

FINISHED ▶ ⌵ 📖 ⚙

📊

📈

📉

📊

📈

📉

⬇️

▼

createdAt
1,485,475,664
1,485,475,665
1,485,475,666
1,485,475,667
1,485,475,668
1,485,475,669
1,485,475,670
1,485,475,671
1.485.475.672

def sentiment(s:String) : String = {

val positive = Array("like", "love", "good", "great", "happy", "cool", "the", "one", "i

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)
        "positive"
    else if(st<0)
        "negative"
    else
        "neutral"
}

// 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.

```

```

sentiment: (s: String)String
res15: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function
1>,StringType,Some(List(StringType)))

```

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

UDF(text)

neutral
negative
positive

```
%sql select sentiment(text), count(1) from tweets where text like '%Morgan Stanley%' group
```

UDF(text)

neutral

%sql select sentiment(text), count(1) from tweets where text like '%NEW PRESIDENT%' group by

FINISHED

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

FINISHED

UDF(text)

neutral

negative

