**Course: CS523 – Big Data Technology  
Prof. Mrudula Mukadam**

**Students**

**Binh Tran 986648**

**Thao Dao 986646**

**Spark Streaming Project**

**(See How to build and run in the last page)**

1. **Project idea**

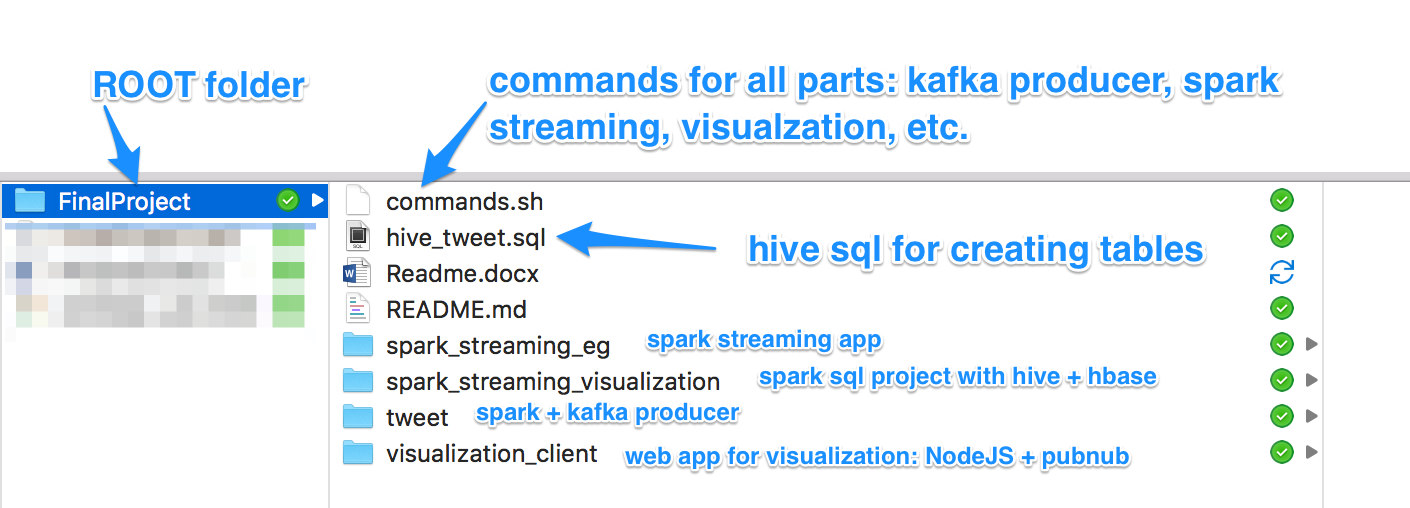
Twitter is known as social network. This project aims to get some insights from twitter feeds for researching purposes

1. **Data set**

Twitter API platform offers options for streaming their real-time Tweets.

1. **Source code and technology**
   1. **Source code**

Unzip the attachment, under root folder you can find commands, HiveQL script and source codes of all applications

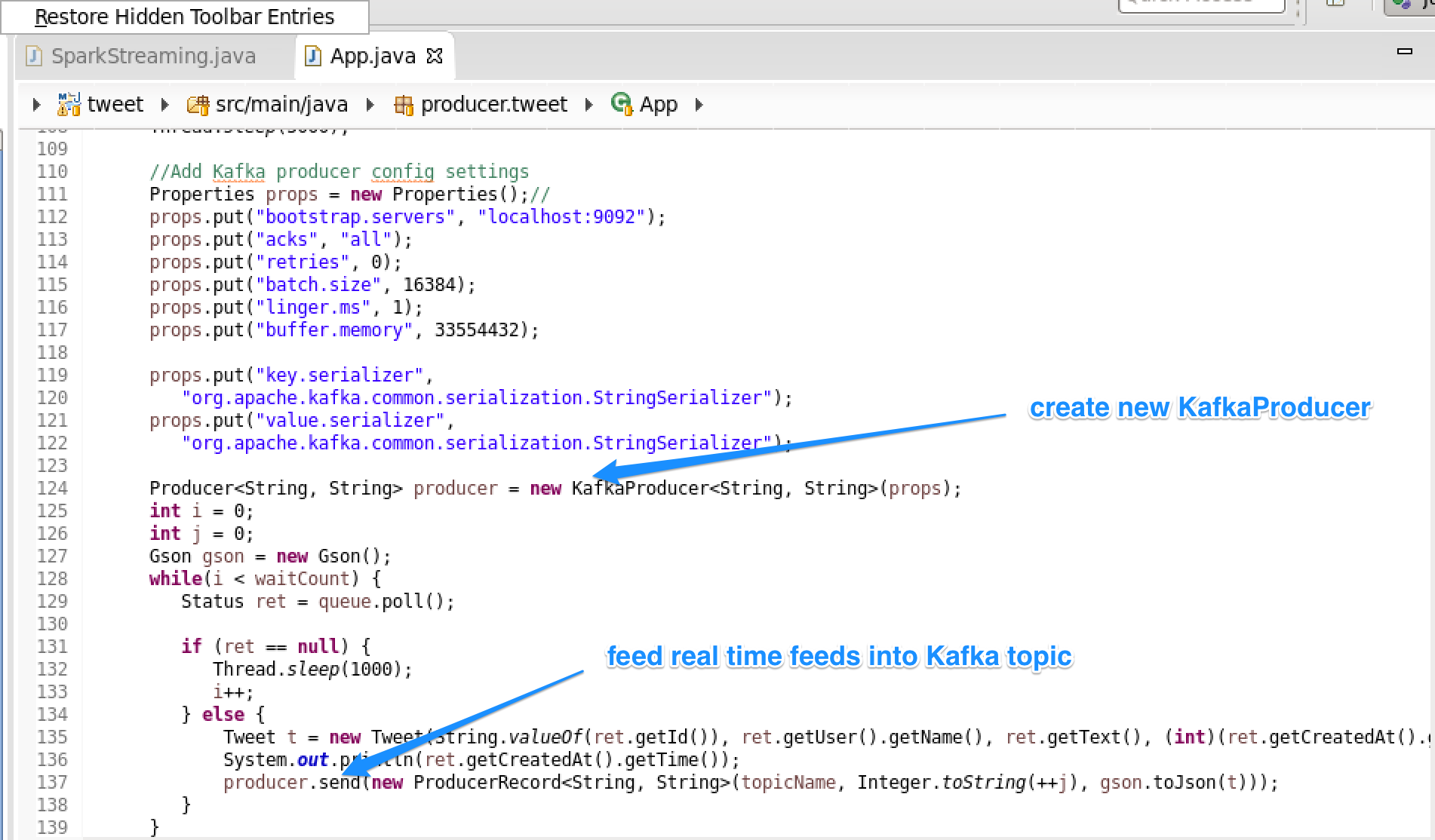


You can also get the source code from github.com at <https://github.com/binhtv/twitter-streaming> for the latest updates

* **Commands detail** (please see how to build and run part)
* **HiveQL script** (please see how to build and run part)
* **tweet folder** (tweet app)

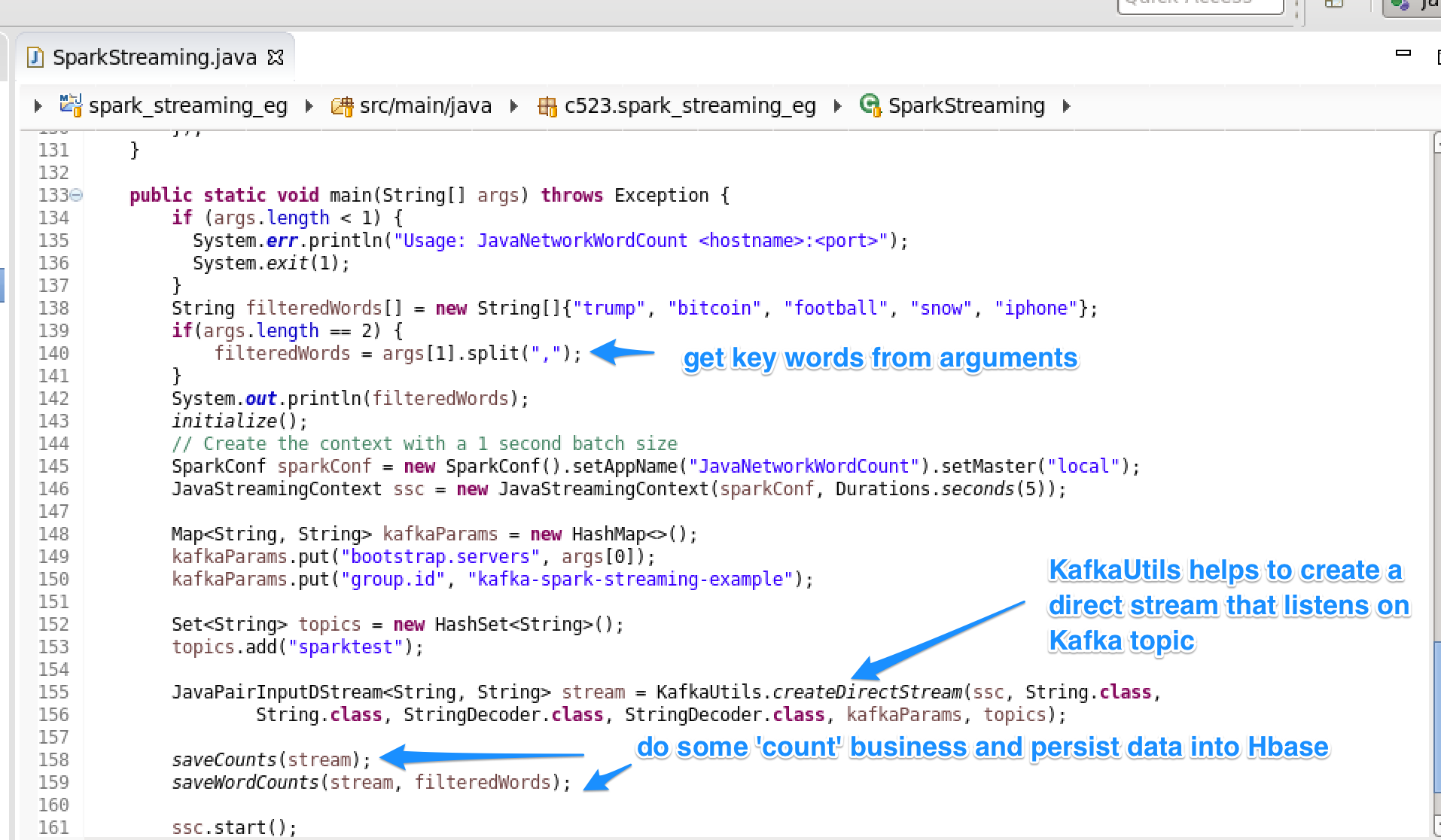
Read real time tweets from twitter by using [Twitter4J](http://twitter4j.org/en/) library

[Apache Kafka](https://kafka.apache.org/) producer take the tweets to feed into Kafka topic



* **spark\_streaming\_eg folder**: contains source code for streaming jobs

Using KafkaUtils to create a direct stream, subscribe to Kafka topic for getting data from Kafka producter (tweet app)



* **spark\_streaming\_visualization folder**

Contains source code for getting data from Hbase (with Hive table on top) for visualization by using Spark SQL

Then, data is broadcasted to web client app for visualization

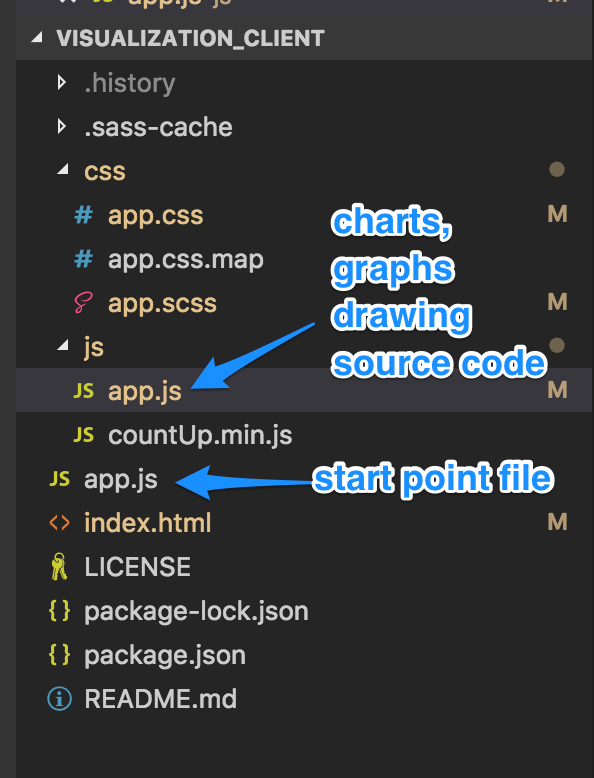


* **visualization\_client folder**

Contain source code for running web client app for visualization. This is a simple web application using NodeJs, Pubnub and Highchart

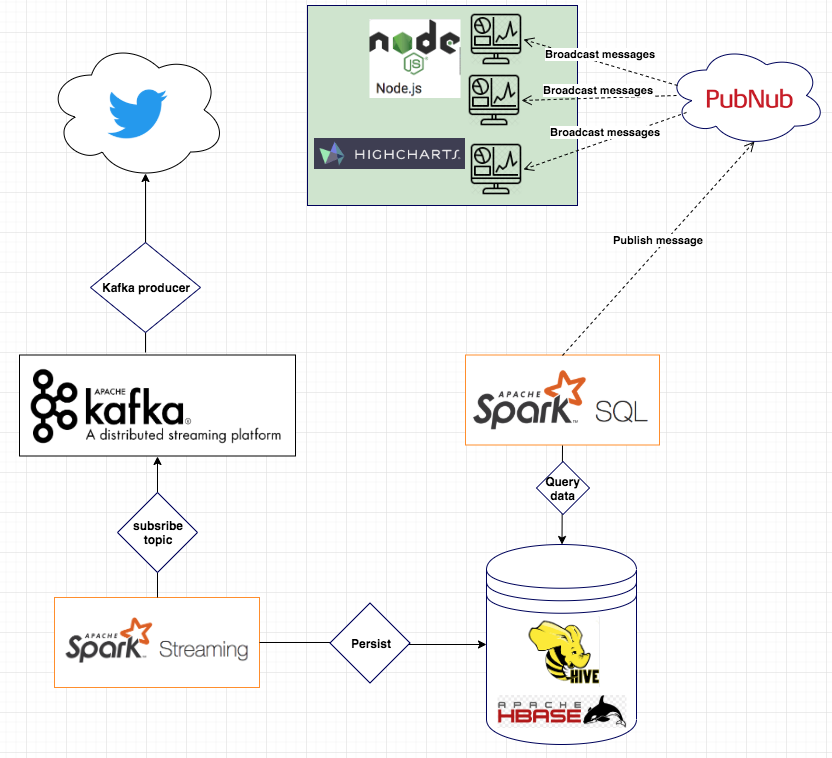
Pubnub subscribe to a channel to get data from this channel

Highchart is javascript library for visualization data with various type of charts and graphs



*Source code structure*

* 1. **Components and flows**

****

*Components and flows*

1. **How to build and run**

**Make sure your latest Kafka service installed**, you can download at <https://kafka.apache.org/> and start manually

**You also need to install zookeeper service**, or you can skip this step if you use *zookeeper* from another service such as HBase or Hadoop

**Make sure Hadoop, Hbase (master & region server), Hive are installed** and working properly in your machine

**Make sure latest NodeJs is installed** on your machine

**Step by step to run:**

* 1. In Kafka folder, Start Kafka service

*$ bin/kafka-server-start.sh config/server.properties*

* 1. Create a Kafka topic

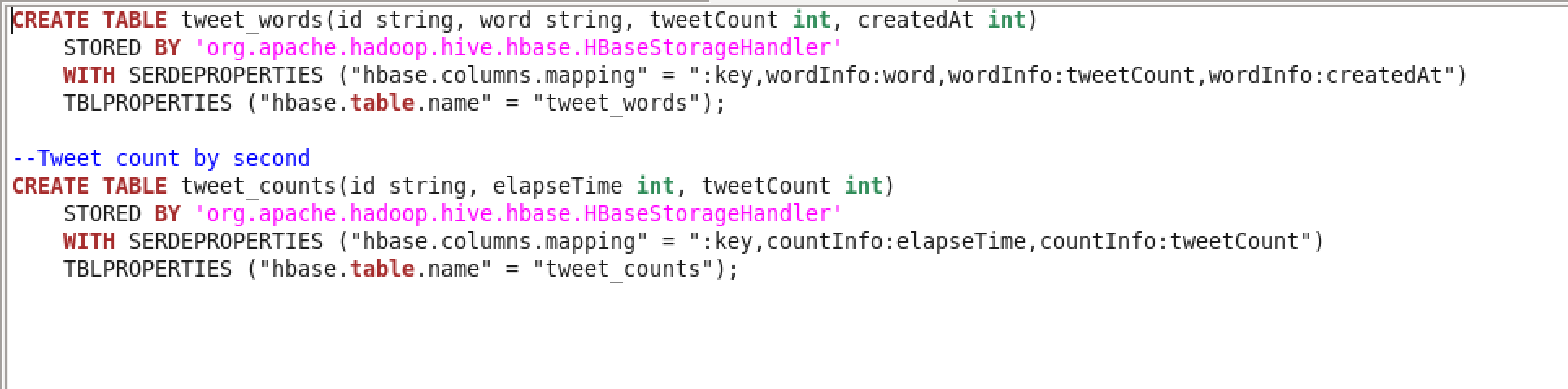
*$ bin/kafka-topics.sh --create --topic sparktest --partitions 1 --replication-factor 1 --zookeeper localhost:2181*

* 1. Copy *hive-site* configuration file: Make hive and spark can work together

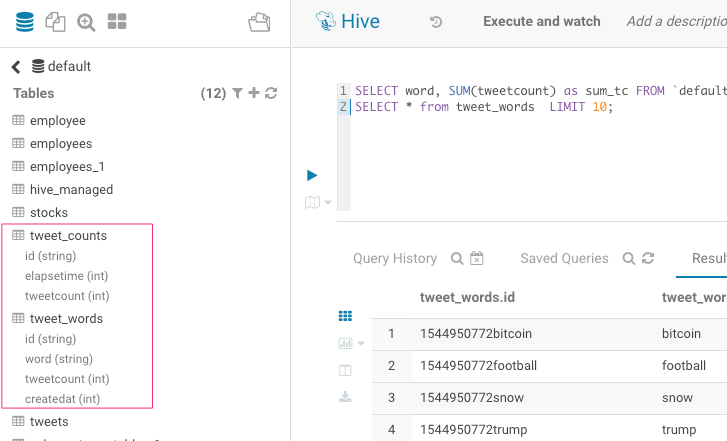
*$ sudo cp /usr/lib/hive/conf/hive-site.xml /usr/lib/spark/conf/*

* 1. At hive shell, create hive tables*: tweet\_counts and tweet\_words*

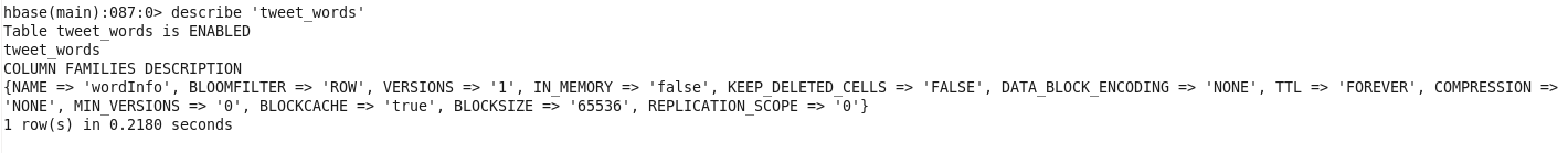
Execute the following scripts in *hive\_tweet.sql*

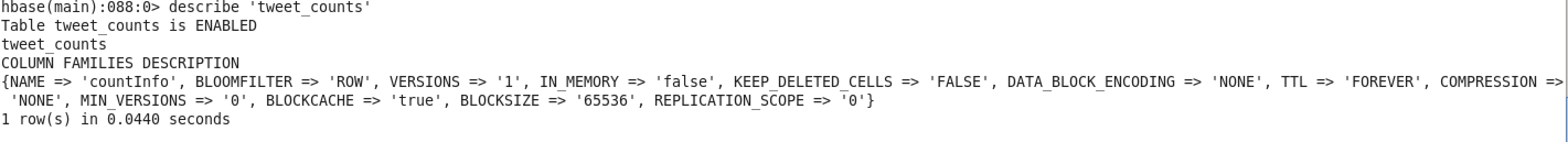
**

After running you should see as the following in Hue

**

From *hbase shell*, you can also see the description of tables by *describe ‘table\_name’*

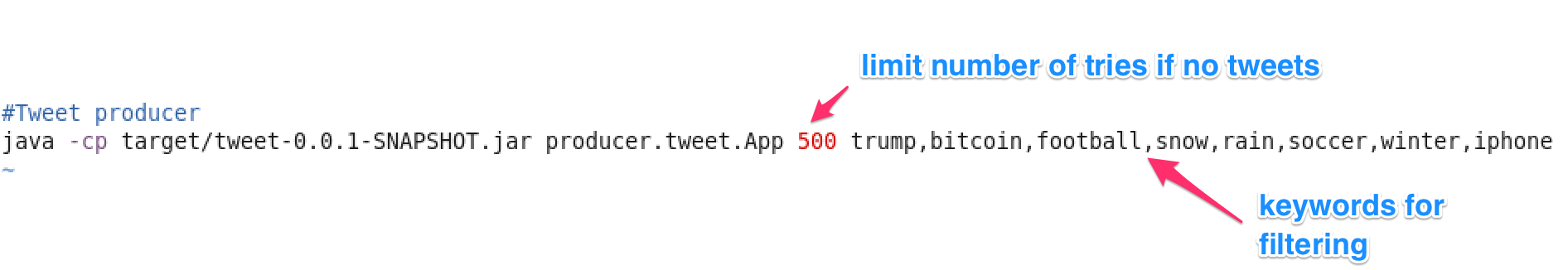
**

**

* 1. At ROOT source code folder, go inside ***tweet***folder

$ *mvn clean install*

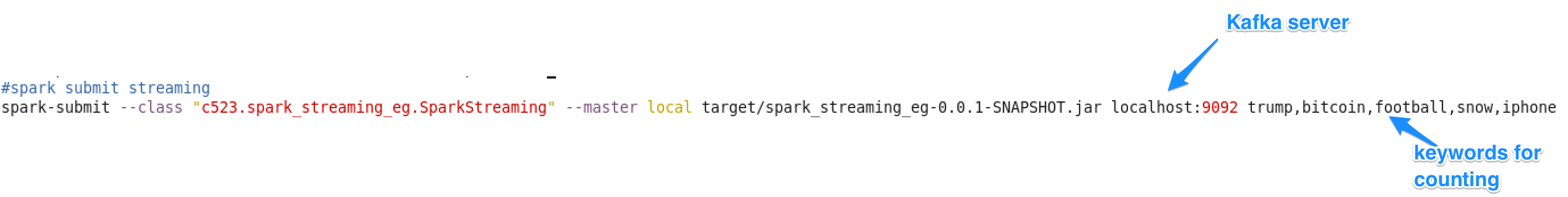
*$ java -cp target/tweet-0.0.1-SNAPSHOT.jar producer.tweet.App 500 trump,bitcoin,football,snow,rain,soccer,winter,iphone*

**

* 1. At ROOT source code folder, go inside ***spark\_streaming\_eg*** folder

$ *mvn clean install*

$ *spark-submit --class "c523.spark\_streaming\_eg.SparkStreaming" --master local target/spark\_streaming\_eg-0.0.1-SNAPSHOT.jar localhost:9092 trump,bitcoin,football,snow,iphone*

**

* 1. At ROOT source code folder, go inside ***spark\_streaming\_visualization***folder

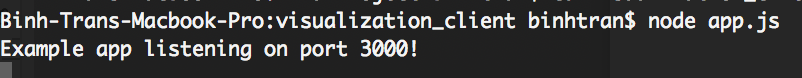
$ *mvn clean install*

$ *spark-submit --class "c523.spark\_streaming\_eg.SparkSql" --master local target/spark\_streaming\_visualization-0.0.1-SNAPSHOT.jar*



* 1. At ROOT source code folder, go inside ***visualization\_client***folder

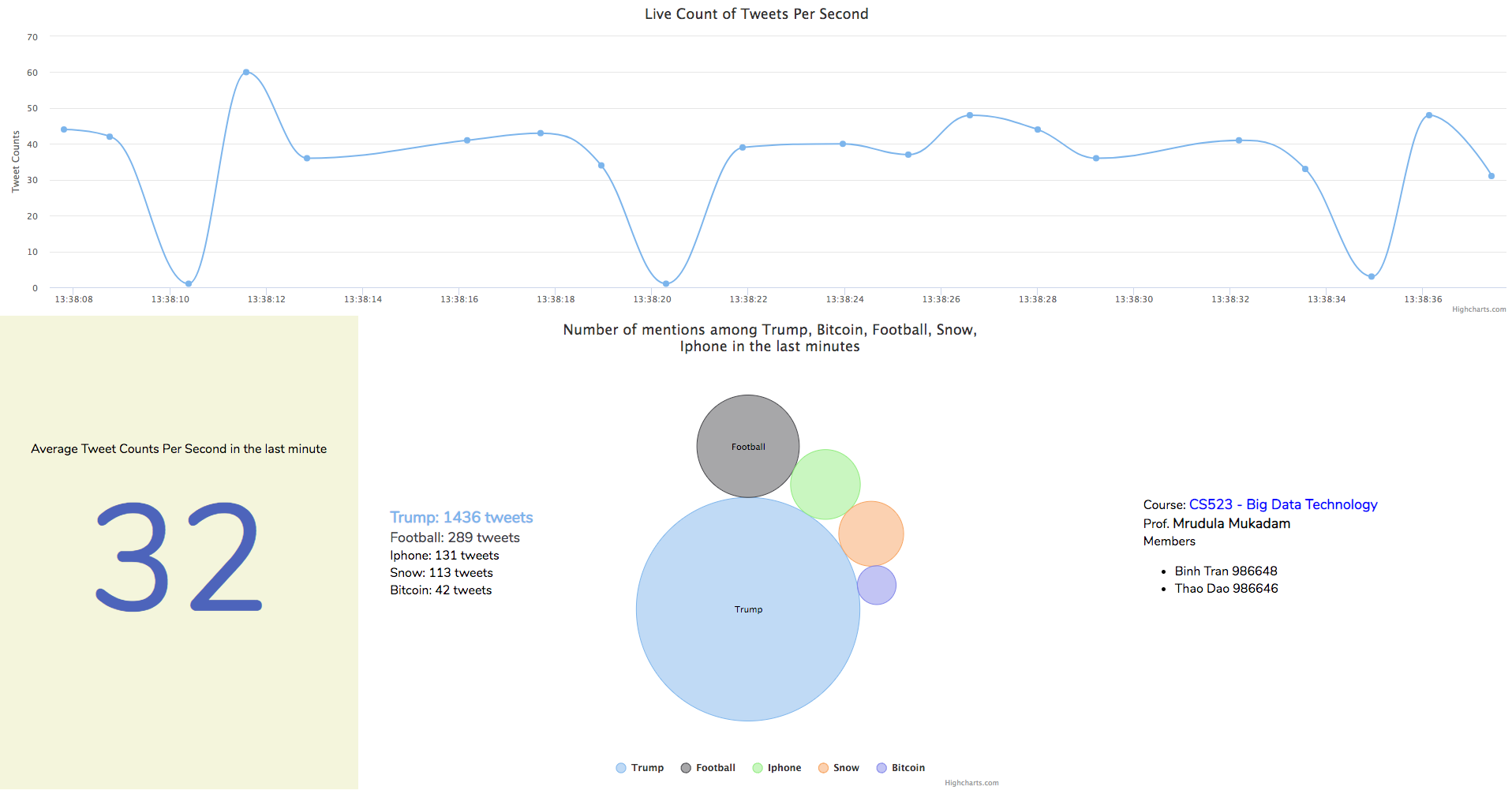
*$ node app.js*



**Test your work**

From browser (Chrome, Firefox, Safari, new IE) go to <http://localhost:3000>

You should see as below:

****