Spark Streaming Basics

# Exercise 1

This exercise is about extracting some knowledge from web-log entries constantly receiving from source.

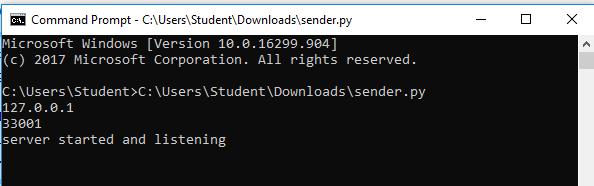
Web-log is simplified for training purpose. Each entry is a line consisting of fields separated by space character.

Sample log is:

172.32.11.121 736443 /api/2.3  
98.134.54.184 182294 /api/2.3  
172.32.11.121 736443 /api/2.3/security  
203.44.122.18 397832 /api/3.0  
172.32.11.121 736443 /api/2.3  
98.134.54.184 182294 /api/3.0  
172.32.11.121 736443 /api/2.3/logging  
172.32.11.121 736443 /api/2.3  
71.11.139.205 291187 /api/2.3  
172.32.11.121 736443 /api/3.0/security  
98.134.54.184 182294 /api/2.3  
71.11.139.205 397832 /api/3.0/logging

First field is IP-address, second one is user id and last one is path to resource.

1. Start nc utility in separate terminal window. It will be sending log entries to localhost on port 33001  
   nc -kl 33001
2. Develop script which receives data from localhost on port 33001 and finds how many times distinct API version is requested and print it on a screen.  
   When code is ready run it in spark shell. Paste lines from web.log file into terminal window where nc is running.  
   Expected output is:  
   (2.3,8)  
   (3.0,4)
3. Additionally, find users who made requests from more than one unique IP-address and print on screen.  
   Expected output is:  
   (397832,2)



This is as far as I got with the task. My IntelliJ would not import any libraries.

This is the bit of code you provided:

**class** another\_streamer {  
 **def** main(args: Array[String]): Unit = {  
 **val** conf = **new** SparkConf().setAppName("another\_streamer").setMaster("local[2]")  
 **val** ssc = **new** StreamingContext(conf, *Seconds*(4))  
  
 **val** requests = ssc.socketTextStream("localhost", 33001)  
  
 **val** columns = requests.map(\_.split(" "))

So this is what I would do, based on what I found in internets:

**val** pairs **=** columns.map(x **=>** (x, 1))

**val** x\_counts **=** pairs.reduceByKey(**\_** + **\_**)

ssc.start()

ssc.awaitTermination()

this should yield the answer for second point.

Now the third point:

Filter out first two columns and reducebykey