

MOONSHOT 2

Spark Enabled Real-time Streaming System

TEAM: ZOOTOPIA

TEAM MEMBERS:

1. Sanjeedha Sanofer Raja (010698805)
2. Meera Mali (010706579)
3. Ojas Milind Kale (010734672)
4. Diksha jain (010711116)
5. Shagun Juneja (010742875)

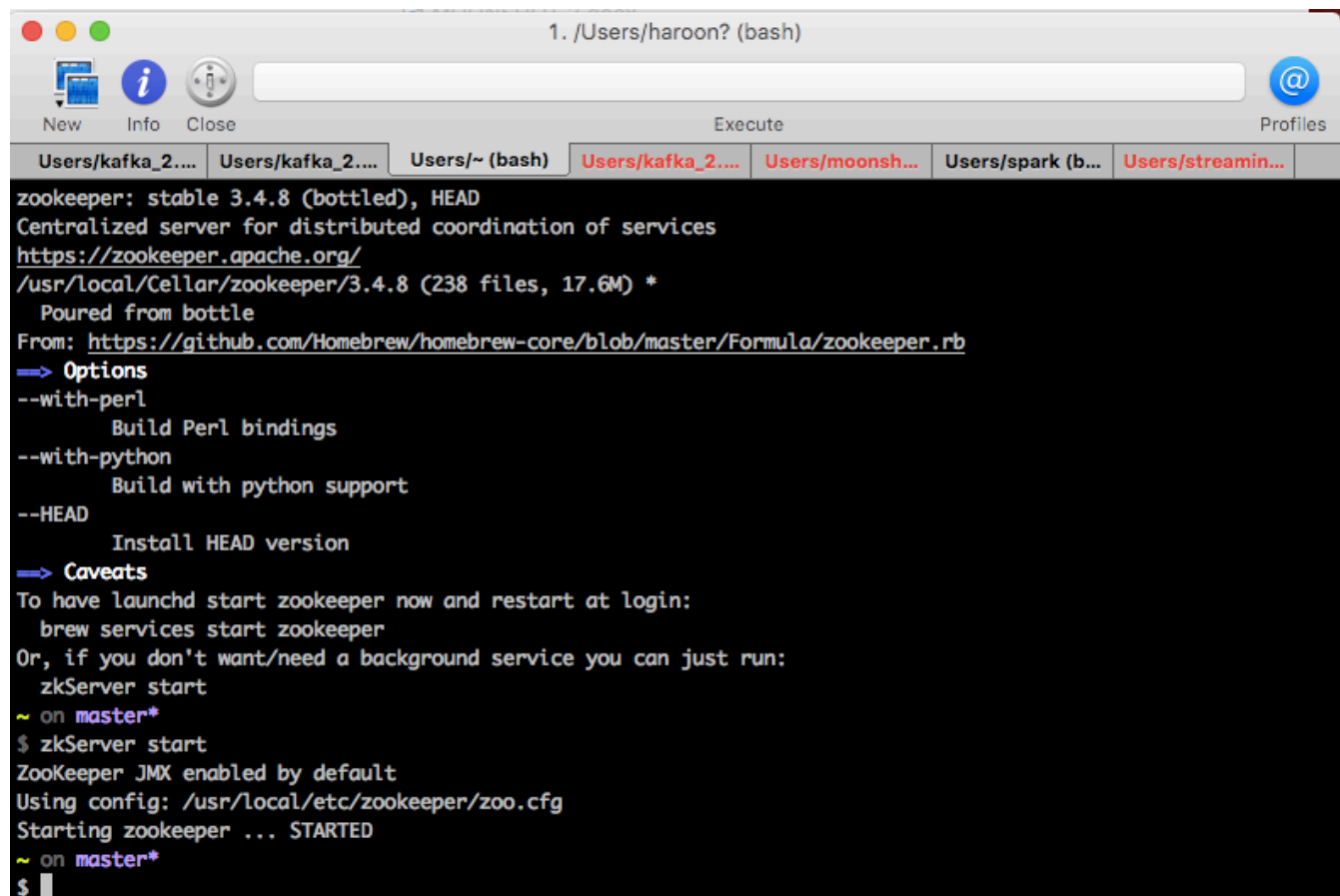
Name of the App: **TweetOViz**

Real-time Streaming data: **Republicans vs Democrats**

Steps:

- 1) Create Twitter API account to get the API key and secret.
- 2) Start Zookeeper installed using brew

zkServer start

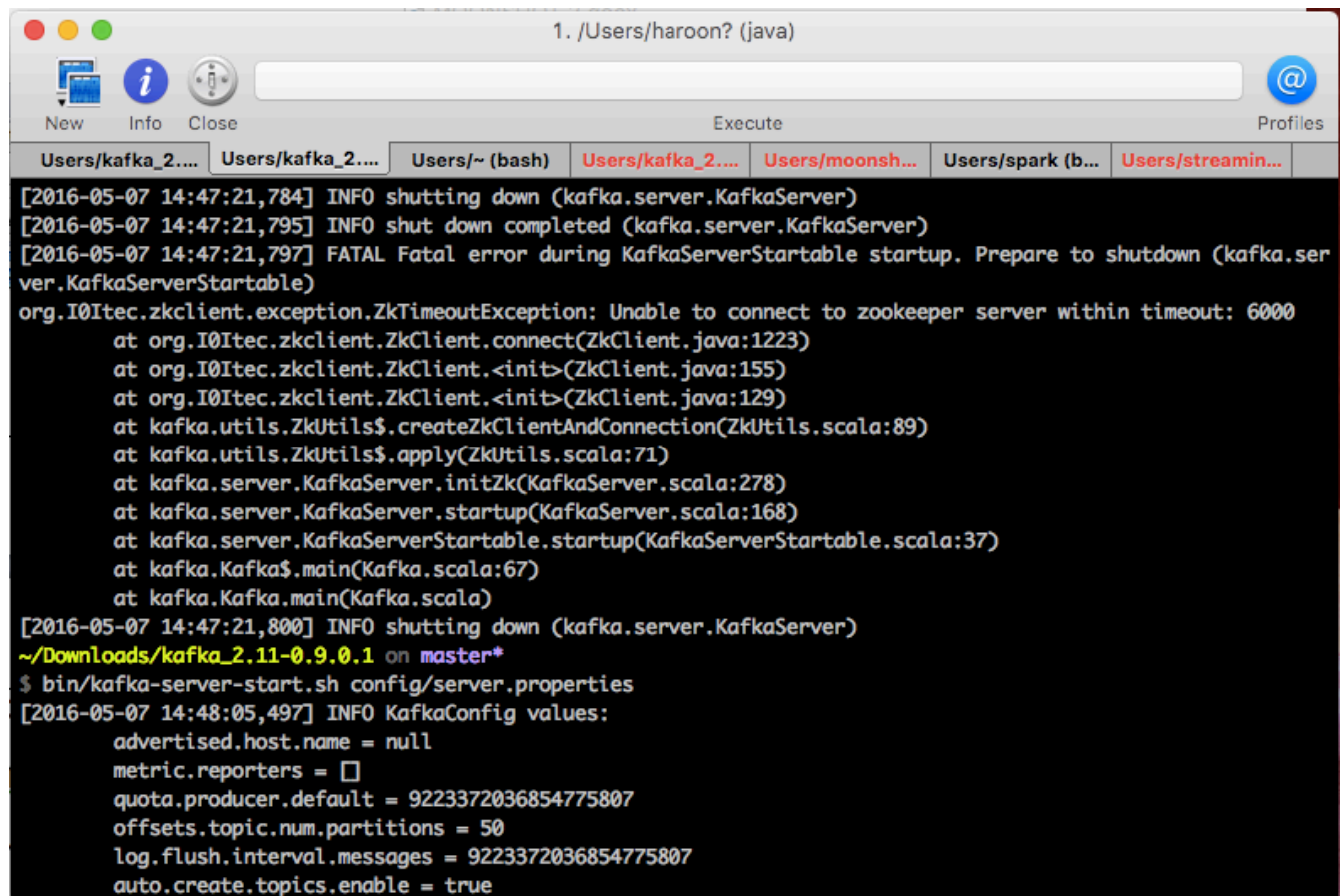


```
1. /Users/haroon? (bash)

zookeeper: stable 3.4.8 (bottled), HEAD
Centralized server for distributed coordination of services
https://zookeeper.apache.org/
/usr/local/Cellar/zookeeper/3.4.8 (238 files, 17.6M) *
  Poured from bottle
From: https://github.com/Homebrew/homebrew-core/blob/master/Formula/zookeeper.rb
-> Options
--with-perl
    Build Perl bindings
--with-python
    Build with python support
--HEAD
    Install HEAD version
-> Caveats
To have launchd start zookeeper now and restart at login:
  brew services start zookeeper
Or, if you don't want/need a background service you can just run:
  zkServer start
~ on master*
$ zkServer start
ZooKeeper JMX enabled by default
Using config: /usr/local/etc/zookeeper/zoo.cfg
Starting zookeeper ... STARTED
~ on master*
$
```

3) Start Kafka

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



The screenshot shows a terminal window titled "1. /Users/haroon? (java)". The window has a menu bar with "New", "Info", "Close", "Execute", and "Profiles". Below the menu bar is a tab bar with several tabs: "Users/kafka_2....", "Users/kafka_2....", "Users/~ (bash)", "Users/kafka_2....", "Users/moonsh...", "Users/spark (b...", and "Users/streamin...". The terminal content shows the following logs and commands:

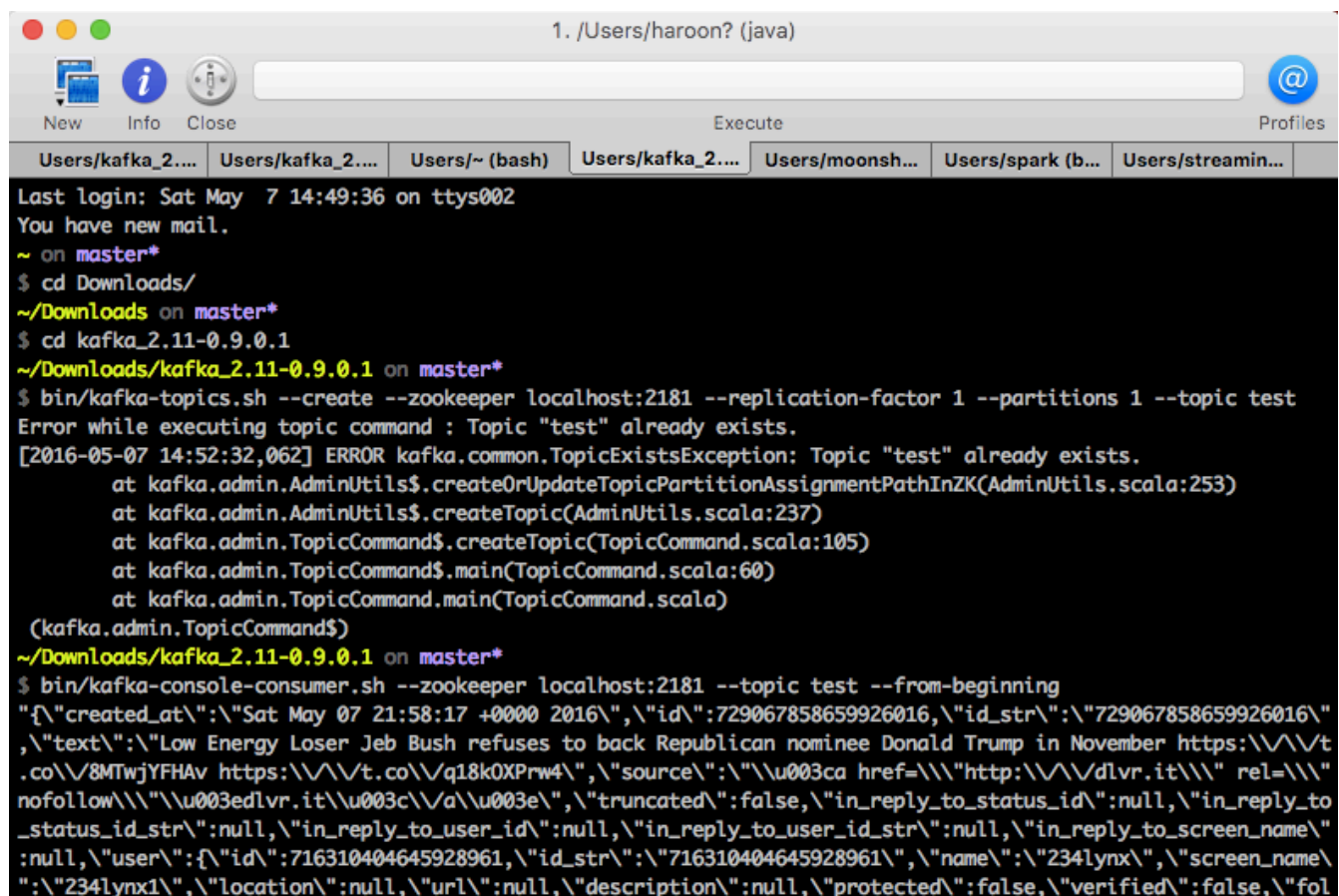
```
[2016-05-07 14:47:21,784] INFO shutting down (kafka.server.KafkaServer)
[2016-05-07 14:47:21,795] INFO shut down completed (kafka.server.KafkaServer)
[2016-05-07 14:47:21,797] FATAL Fatal error during KafkaServerStartable startup. Prepare to shutdown (kafka.server.KafkaServerStartable)
org.I0Itec.zkclient.exception.ZkTimeoutException: Unable to connect to zookeeper server within timeout: 6000
    at org.I0Itec.zkclient.ZkClient.connect(ZkClient.java:1223)
    at org.I0Itec.zkclient.ZkClient.<init>(ZkClient.java:155)
    at org.I0Itec.zkclient.ZkClient.<init>(ZkClient.java:129)
    at kafka.utils.ZkUtils$.createZkClientAndConnection(ZkUtils.scala:89)
    at kafka.utils.ZkUtils$.apply(ZkUtils.scala:71)
    at kafka.server.KafkaServer.initZk(KafkaServer.scala:278)
    at kafka.server.KafkaServer.startup(KafkaServer.scala:168)
    at kafka.server.KafkaServerStartable.startup(KafkaServerStartable.scala:37)
    at kafka.Kafka$.main(Kafka.scala:67)
    at kafka.Kafka.main(Kafka.scala)
[2016-05-07 14:47:21,800] INFO shutting down (kafka.server.KafkaServer)
~/Downloads/kafka_2.11-0.9.0.1 on master*
$ bin/kafka-server-start.sh config/server.properties
[2016-05-07 14:48:05,497] INFO KafkaConfig values:
    advertised.host.name = null
    metric.reporters = []
    quota.producer.default = 9223372036854775807
    offsets.topic.num.partitions = 50
    log.flush.interval.messages = 9223372036854775807
    auto.create.topics.enable = true
```

4) Create Kafka topic if necessary

```
bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test
```

5) Create kafka Consumer

```
bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic test --from-beginning
```



The screenshot shows a terminal window titled "1. /Users/haroon? (java)". The terminal output shows the following sequence of events:

- System messages: "Last login: Sat May 7 14:49:36 on ttys002" and "You have new mail."
- User prompt: "~ on master*" (where master is highlighted in yellow)
- Command: `$ cd Downloads/`
- User prompt: "~/Downloads on master*" (where Downloads is highlighted in yellow)
- Command: `$ cd kafka_2.11-0.9.0.1`
- User prompt: "~/Downloads/kafka_2.11-0.9.0.1 on master*" (where kafka_2.11-0.9.0.1 is highlighted in yellow)
- Command: `$ bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test`
- Error message: "Error while executing topic command : Topic "test" already exists."
- Stack trace: `[2016-05-07 14:52:32,062] ERROR kafka.common.TopicExistsException: Topic "test" already exists. at kafka.admin.AdminUtils$.createOrUpdateTopicPartitionAssignmentPathInZK(AdminUtils.scala:253) at kafka.admin.AdminUtils$.createTopic(AdminUtils.scala:237) at kafka.admin.TopicCommand$.createTopic(TopicCommand.scala:105) at kafka.admin.TopicCommand$.main(TopicCommand.scala:60) at kafka.admin.TopicCommand.main(TopicCommand.scala)`
- User prompt: "(kafka.admin.TopicCommand\$)"
- User prompt: "~/Downloads/kafka_2.11-0.9.0.1 on master*" (where kafka_2.11-0.9.0.1 is highlighted in yellow)
- Command: `$ bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic test --from-beginning`
- Output (a JSON tweet): `{"created_at":"Sat May 07 21:58:17 +0000 2016","id":729067858659926016,"id_str":"729067858659926016","text":"Low Energy Loser Jeb Bush refuses to back Republican nominee Donald Trump in November https://t.co/8MTwjYFHA https://t.co/q18k0XPw4","source":"\u003ca href=\"http://d1vr.it/\" rel=\"nofollow\" \u003cd1vr.it\u003c/a\u003e","truncated":false,"in_reply_to_status_id":null,"in_reply_to_status_id_str":null,"in_reply_to_user_id":null,"in_reply_to_user_id_str":null,"in_reply_to_screen_name":null,"user":{"id":716310404645928961,"id_str":"716310404645928961","name":"234lynx","screen_name":"234lynx1","location":null,"url":null,"description":null,"protected":false,"verified":false,"fol`

6) Run the python script to use twitter API and get tweets and send it to kafka broker

```
python twitter_streaming.py
```

twitter_streaming.py

```
#Import the necessary methods from tweepy library
from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream
from kafka import KafkaProducer
import json

producer = KafkaProducer(value_serializer=lambda v: json.dumps(v).encode('utf-8'))

#Variables that contains the user credentials to access Twitter API
access_token = "4176278119-OkdJ TZgtdJh6qZYz483HTdMjcDAVbJQL45CoSgZ"
access_token_secret = "Al6PWC6EXWNPUvvmP964kVDfZivlCyVEEdIUaCQYoecxx"
consumer_key = "j19KmpBmWHDZVbORuUgXj3aMF"
consumer_secret =
"MrzG1HCftg9lgEu93QMnjtPfo4AxFb8YkzEXe4HvZZGbFcm4NO"

#This is a basic listener that just prints received tweets to stdout.
class StdOutListener(StreamListener):

    def on_data(self, data):
        producer.send('test', data)
        return True

    def on_error(self, status):
        print status

if __name__ == '__main__':

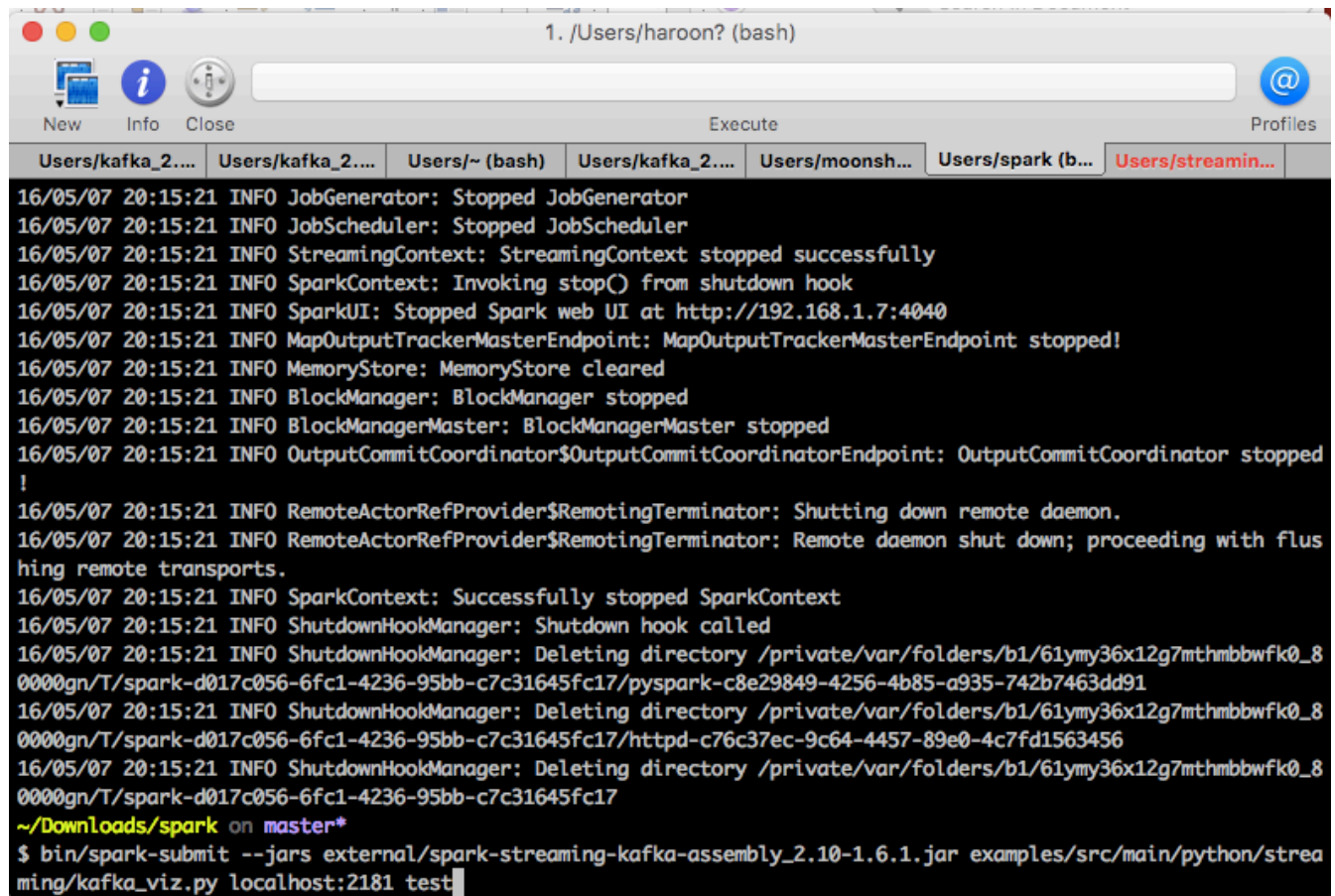
    #This handles Twitter authentication and the connection to Twitter Streaming
    API
    l = StdOutListener()
    auth = OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    stream = Stream(auth, l)

    #This line filter Twitter Streams to capture data by the keywords: 'python',
    'javascript', 'ruby'
```

```
stream.filter(track=['democrat', 'republican'])
```

7) Run the spark kafka script to listen to kafka broker, receive the tweets from kafka, process the tweets and count the number of tweets for democrats and republicans. Save the information in a JSON file.

```
bin/spark-submit --jars external/spark-streaming-kafka-assembly_2.10-1.6.1.jar  
examples/src/main/python/streaming/kafka_viz.py localhost:2181 test
```

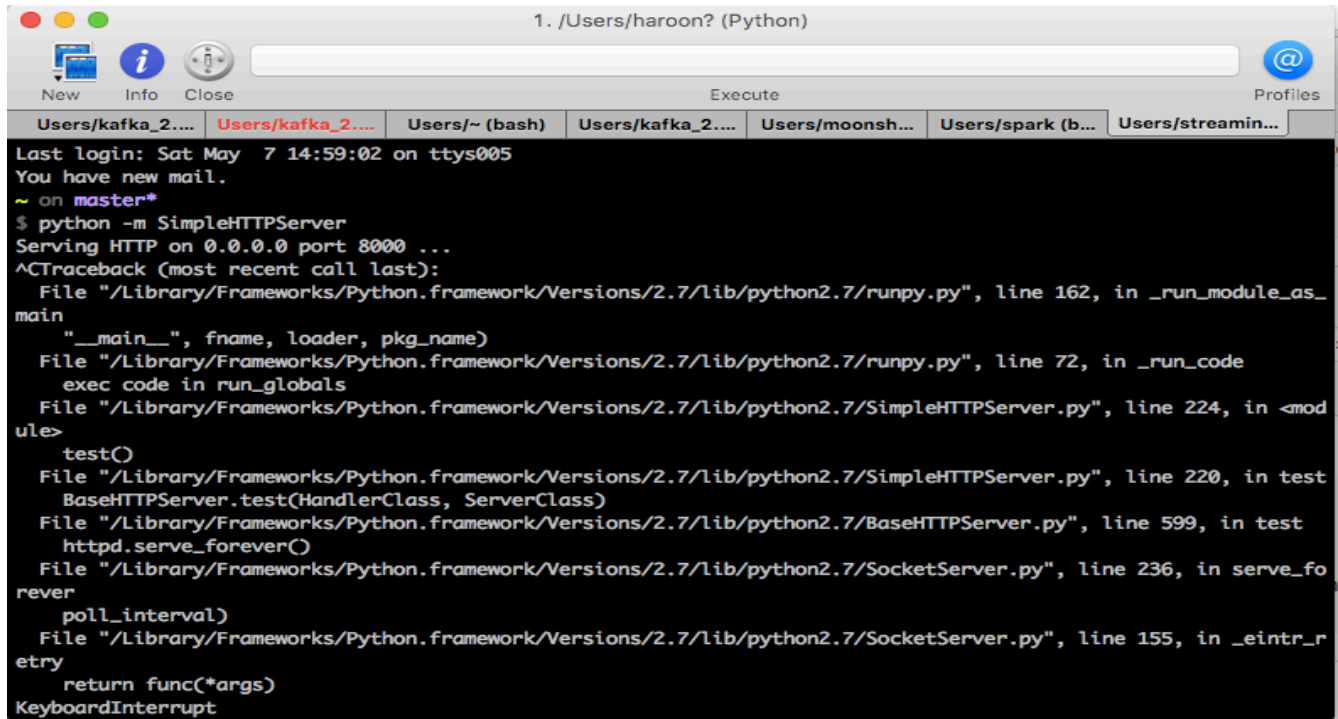


The screenshot shows a terminal window titled "1. /Users/haroon? (bash)". The terminal displays a series of log messages from Spark, indicating the successful shutdown of various components like JobGenerator, JobScheduler, StreamingContext, SparkContext, SparkUI, MapOutputTrackerMasterEndpoint, MemoryStore, BlockManager, BlockManagerMaster, OutputCommitCoordinator, and RemoteActorRefProvider. The logs are timestamped "16/05/07 20:15:21". At the bottom of the terminal, the command `bin/spark-submit --jars external/spark-streaming-kafka-assembly_2.10-1.6.1.jar examples/src/main/python/streaming/kafka_viz.py localhost:2181 test` is entered.

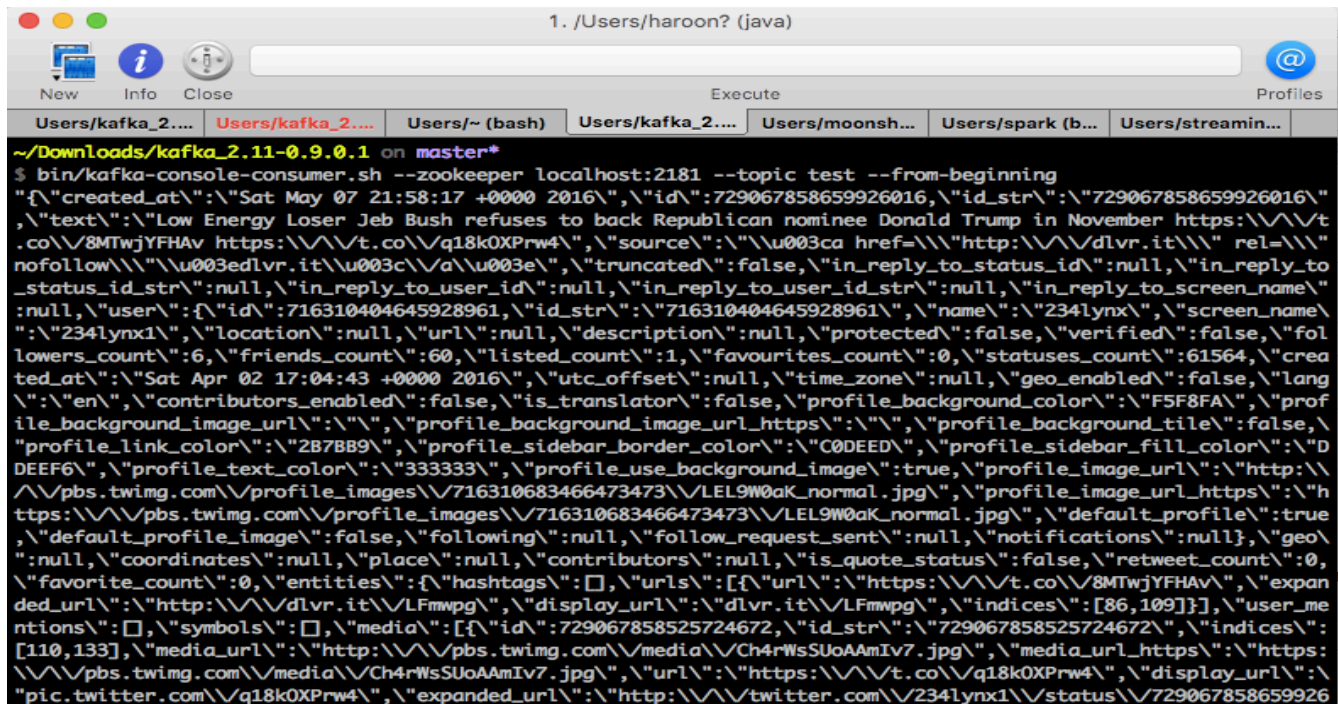
```
16/05/07 20:15:21 INFO JobGenerator: Stopped JobGenerator
16/05/07 20:15:21 INFO JobScheduler: Stopped JobScheduler
16/05/07 20:15:21 INFO StreamingContext: StreamingContext stopped successfully
16/05/07 20:15:21 INFO SparkContext: Invoking stop() from shutdown hook
16/05/07 20:15:21 INFO SparkUI: Stopped Spark web UI at http://192.168.1.7:4040
16/05/07 20:15:21 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
16/05/07 20:15:21 INFO MemoryStore: MemoryStore cleared
16/05/07 20:15:21 INFO BlockManager: BlockManager stopped
16/05/07 20:15:21 INFO BlockManagerMaster: BlockManagerMaster stopped
16/05/07 20:15:21 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped
!
16/05/07 20:15:21 INFO RemoteActorRefProvider$RemotingTerminator: Shutting down remote daemon.
16/05/07 20:15:21 INFO RemoteActorRefProvider$RemotingTerminator: Remote daemon shut down; proceeding with flushing remote transports.
16/05/07 20:15:21 INFO SparkContext: Successfully stopped SparkContext
16/05/07 20:15:21 INFO ShutdownHookManager: Shutdown hook called
16/05/07 20:15:21 INFO ShutdownHookManager: Deleting directory /private/var/folders/b1/61my36x12g7mthmbbwfk0_80000gn/T/spark-d017c056-6fc1-4236-95bb-c7c31645fc17/pyspark-c8e29849-4256-4b85-a935-742b7463dd91
16/05/07 20:15:21 INFO ShutdownHookManager: Deleting directory /private/var/folders/b1/61my36x12g7mthmbbwfk0_80000gn/T/spark-d017c056-6fc1-4236-95bb-c7c31645fc17/httpd-c76c37ec-9c64-4457-89e0-4c7fd1563456
16/05/07 20:15:21 INFO ShutdownHookManager: Deleting directory /private/var/folders/b1/61my36x12g7mthmbbwfk0_80000gn/T/spark-d017c056-6fc1-4236-95bb-c7c31645fc17
~/Downloads/spark on master*
$ bin/spark-submit --jars external/spark-streaming-kafka-assembly_2.10-1.6.1.jar examples/src/main/python/streaming/kafka_viz.py localhost:2181 test
```


8) Use d3.js to read the JSON file in step 7 at regular intervals and plot the data in a bar chart and time series chart.

python -m SimpleHTTPServer



```
1. /Users/haroon? (Python)
Last login: Sat May 7 14:59:02 on ttys005
You have new mail.
~ on master*
$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
^CTraceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/runpy.py", line 162, in _run_module_as_main
    "__main__", fname, loader, pkg_name)
  File "/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/runpy.py", line 72, in _run_code
    exec code in run_globals
  File "/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/SimpleHTTPServer.py", line 224, in <module>
    test()
  File "/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/SimpleHTTPServer.py", line 220, in test
    BaseHTTPServer.test(HandlerClass, ServerClass)
  File "/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/BaseHTTPServer.py", line 599, in test
    httpd.serve_forever()
  File "/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/SocketServer.py", line 236, in serve_forever
    poll_interval)
  File "/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/SocketServer.py", line 155, in _eternity
    return func(*args)
KeyboardInterrupt
```



```
1. /Users/haroon? (java)
~/Downloads/kafka_2.11-0.9.0.1 on master*
$ bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic test --from-beginning
{"created_at":"Sat May 07 21:58:17 +0000 2016","id":"729067858659926016","id_str":"729067858659926016","text":"Low Energy Loser Jeb Bush refuses to back Republican nominee Donald Trump in November https://t.co/8MTwjYFHAv https://t.co/q18k0XPw4","source":{"url":"http://dvr.it/1LFmwpG","rel":"nofollow","truncated":false,"in_reply_to_status_id":null,"in_reply_to_status_id_str":null,"in_reply_to_user_id":null,"in_reply_to_user_id_str":null,"in_reply_to_screen_name":null,"user":{"id":"716310404645928961","id_str":"716310404645928961","name":"234lynx","screen_name":"234lynx1","location":null,"url":null,"description":null,"protected":false,"verified":false,"followers_count":6,"friends_count":60,"listed_count":1,"favourites_count":0,"statuses_count":61564,"created_at":"Sat Apr 02 17:04:43 +0000 2016","utc_offset":null,"time_zone":null,"geo_enabled":false,"lang":"en","contributors_enabled":false,"is_translator":false,"profile_background_color":"F5F8FA","profile_background_image_url":"","profile_background_image_url_https":"","profile_background_tile":false,"profile_link_color":"2B78B9","profile_sidebar_border_color":"C0DEED","profile_sidebar_fill_color":"DDEEFF","profile_text_color":"333333","profile_use_background_image":true,"profile_image_url":"http://pbs.twimg.com/profile_images/716310683466473473/LEL9W0aK_normal.jpg","profile_image_url_https":"https://pbs.twimg.com/profile_images/716310683466473473/LEL9W0aK_normal.jpg","default_profile":true,"default_profile_image":false,"following":null,"follow_request_sent":null,"notifications":null,"geo":{"coordinates":null,"place":null,"contributors":null,"is_quote_status":false,"retweet_count":0,"favorite_count":0,"entities":{"hashtags":[],"urls":[{"url":"https://t.co/8MTwjYFHAv","expanded_url":"http://dvr.it/1LFmwpG","display_url":"dvr.it/1LFmwpG","indices":{"start":86,"end":109}}],"user_mentions":[],"symbols":[],"media":[{"id":"729067858525724672","id_str":"729067858525724672","indices":{"start":110,"end":133},"media_url":"http://pbs.twimg.com/media/Ch4rWsSUoAAmIv7.jpg","media_url_https":"https://pbs.twimg.com/media/Ch4rWsSUoAAmIv7.jpg","url":"https://t.co/q18k0XPw4","display_url":"pic.twitter.com/q18k0XPw4","expanded_url":"http://twitter.com/234lynx1/status/729067858659926016"}]}}
```

9) Run the URL for visualizing the flow.

<http://localhost:8000/streaming.html>

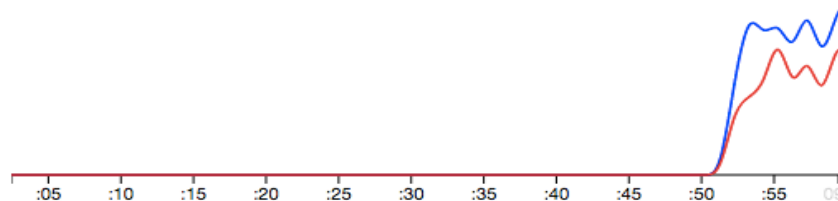
Democrats vs Republican

What Are People Tweeting Right Now?

Number Of Tweets (5 seconds window)



— Democrat — Republican



streaming.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="http://d3js.org/d3.v3.min.js" charset="utf-8"></script>
```

```
<link rel="stylesheet" href="style.css">
```

```
<style type="text/css">
```

```
html, body, div, span, applet, object, iframe,
```

```
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
```

```
a, abbr, acronym, address, big, cite, code,
```

```
del, dfn, em, img, ins, kbd, q, s, samp,
```

```
small, strike, strong, sub, sup, tt, var,
```

```
b, u, i, center,
```

```
dl, dt, dd, ol, ul, li,
```



```
fieldset, form, label, legend,  
table, caption, tbody, tfoot, thead, tr, th, td,  
article, aside, canvas, details, embed,  
figure, figcaption, footer, header, hgroup,  
menu, nav, output, ruby, section, summary,  
time, mark, audio, video {  
    margin: 0;  
    padding: 0;  
    border: 0;  
    font: inherit;  
    font-size: 100%;  
    vertical-align: baseline;  
    font-family: "Helvetica Neue", Helvetica, Arial, "Lucida Grande", sans-serif;  
    font-weight: inherit;  
}
```

```
h1 {  
    font-size: 300%;  
    font-family: "HelveticaNeue-Light", "Helvetica Neue Light", "Helvetica Neue", Helvetica, Arial,  
"Lucida Grande", sans-serif;  
    font-weight: 600;  
    text-align: left;  
    margin-top: 15px;  
    margin-bottom: 15px;  
}
```

```
h2 {  
    font-size: 150%;  
    font-family: "HelveticaNeue-Light", "Helvetica Neue Light", "Helvetica Neue", Helvetica, Arial,  
"Lucida Grande", sans-serif;  
    font-weight: 300;  
    text-align: left;  
    margin-top: 5px;  
    margin-bottom: 5px;  
}
```

```
h3 {  
    font-size: 12px;  
    font-style: italic;  
    color: gray;  
    margin-top: 5px;  
    margin-bottom: 5px;  
}
```

```
#everything{  
    width: 600px;  
    margin-top: 200px;  
    margin-left: auto;  
    margin-right: auto;
```

```
}
#chart{
  width:800px;
  height:200px;
}
.bar{
  fill:#eaeaea;
}
text.label{
  fill: white;
  color: white;
  font-size: 20px;
  font-weight: bold;
}
text.category{
  fill: white;
  font-size: 14px;
}

.graph .axis {
  stroke-width: 1;
}

.graph .axis .tick line {
  stroke: black;
}

.graph .axis .tick text {
  fill: black;
  font-size: 0.7em;
}

.graph .axis .domain {
  fill: none;
  stroke: black;
}

.graph .group {
  fill: none;
  stroke: black;
  stroke-width: 1.5;
}
</style>
</head>

<body>

<div id="everything">
```

```
<h1>Democrats vs Republican</h1>
<h2>What Are People Tweeting Right Now?</h2>
<h3>Number Of Tweets (5 seconds window)</h3>
<div id="chart"></div>
```

```

<div class="graph"></div>
</div>
```

```
<script>
```

```
//updatingBarChart.js
```

```
var setup = function(targetID){
  //Set size of svg element and chart
  var margin = {top: 0, right: 0, bottom: 0, left: 0},
    width = 600 - margin.left - margin.right,
    height = 100 - margin.top - margin.bottom,
    categoryIndent = 4*15 + 5,
    defaultBarWidth = 2000;

  //Set up scales
  var x = d3.scale.linear()
    .domain([0,defaultBarWidth])
    .range([0,width]);

  var y = d3.scale.ordinal()
    .rangeRoundBands([0, height], 0.1, 0);

  //Create SVG element
  d3.select(targetID).selectAll("svg").remove()
  var svg = d3.select(targetID).append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
    .append("g")
    .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

  //Package and export settings
  var settings = {
    margin:margin, width:width, height:height, categoryIndent:categoryIndent,
    svg:svg, x:x, y:y
  }
  return settings;
}

var redrawChart = function(targetID, newdata) {
```

```

//Import settings
var margin=settings.margin, width=settings.width, height=settings.height,
categoryIndent=settings.categoryIndent,
svg=settings.svg, x=settings.x, y=settings.y;

//Reset domains
y.domain(newdata.sort(function(a,b){
  return b.value - a.value;
}))
.map(function(d) { return d.key; });
var barmax = d3.max(newdata, function(e) {
  return e.value;
});
x.domain([0,barmax]);

//////////
//ENTER//
//////////

//Bind new data to chart rows

//Create chart row and move to below the bottom of the chart
var chartRow = svg.selectAll("g.chartRow")
  .data(newdata, function(d){ return d.key});
var newRow = chartRow
  .enter()
  .append("g")
  .attr("class", "chartRow")
  .attr("transform", "translate(0," + height + margin.top + margin.bottom + ")");

//Add rectangles
newRow.insert("rect")
  .attr("class","bar")
  .attr("x", 0)
  .attr("opacity",0)
  .attr("height", y.rangeBand())
  .style("fill", function(d) { return d.color; })
  .attr("width", function(d) { return x(d.value);})

//Add value labels
newRow.append("text")
  .attr("class","label")
  .attr("y", y.rangeBand()/2)
  .attr("x",0)
  .attr("opacity",0)
  .attr("dy", ".35em")
  .attr("dx", "0.5em")
  .text(function(d){return d.value;});

```

```

//Add Headlines
newRow.append("text")
  .attr("class","category")
  .attr("text-overflow","ellipsis")
  .attr("y", y.rangeBand()/2)
  .attr("x",categoryIndent)
  .attr("opacity",0)
  .attr("dy",".35em")
  .attr("dx","0.5em")
  .text(function(d){return d.key});

//////////
//UPDATE//
//////////

//Update bar widths
chartRow.select(".bar").transition()
  .duration(300)
  .attr("width", function(d) { return x(d.value);})
  .attr("opacity",1);

//Update data labels
chartRow.select(".label").transition()
  .duration(300)
  .attr("opacity",1)
  .tween("text", function(d) {
    var i = d3.interpolate(+this.textContent.replace(/\./g,""), +d.value);
    return function(t) {
      this.textContent = Math.round(i(t));
    };
  });

//Fade in categories
chartRow.select(".category").transition()
  .duration(300)
  .attr("opacity",1);

//////////
//EXIT//
//////////

//Fade out and remove exit elements
chartRow.exit().transition()
  .style("opacity","0")
  .attr("transform", "translate(0," + (height + margin.top + margin.bottom) + ")")

```

```

.remove();

//////////
//REORDER ROWS//
//////////

var delay = function(d, i) { return 200 + i * 10; };

chartRow.transition()
  .delay(delay)
  .duration(500)
  .attr("transform", function(d){ return "translate(0," + y(d.key) + ")"; });
};

//Pulls data
//Since our data is fake, adds some random changes to simulate a data stream.
//Uses a callback because d3.json loading is asynchronous
var pullData = function(settings,callback){
  d3.json("testdata.json", function (err, data){
    if (err) return console.warn(err);

    var newData = data;
    data.forEach(function(d,i){
      var newValue = d.value + Math.floor((Math.random()*10) - 5)
      newData[i].value = newValue <= 0 ? 10 : newValue
    })

    newData = formatData(newData);

    callback(settings,newData);
  })
}

//Sort data in descending order and take the top 10 values
var formatData = function(data){
  return data.sort(function (a, b) {
    return b.value - a.value;
  })
  .slice(0, 10);
}

//I like to call it what it does
var redraw = function(settings){
  pullData(settings,redrawChart)
}

```

```
//setup (includes first draw)
var settings = setup('#chart');
redraw(settings)

//Repeat every 3 seconds
setInterval(function(){
  redraw(settings)
}, 5000);

</script>

<script src="http://d3js.org/d3.v3.min.js"></script>
  <script>
    var limit = 60 * 1,
        duration = 1000,
        now = new Date(Date.now() - duration)

    var width = 500,
        height = 200

    var groups = {
      democrat: {
        value: 0,
        color: 'blue',
        data: d3.range(limit).map(function() {
          return 0
        })
      },
      republican: {
        value: 0,
        color: 'red',
        data: d3.range(limit).map(function() {
          return 0
        })
      }
    }
  }

  var xx = d3.time.scale()
    .domain([now - (limit - 2), now - duration])
    .range([0, width])

  var yy = d3.scale.linear()
    .domain([0, 30])
    .range([height, 0])

  var line = d3.svg.line()
    .interpolate("basis")
```



```

.x(function(d, i) {
    return xx(now - (limit - 1 - i) * duration)
})
.y(function(d) {
    return yy(d)
})

var svg2 = d3.select('.graph').append('svg')
    .attr('class', 'chart')
    .attr('width', width)
    .attr('height', height + 50)

var axis = svg2.append('g')
    .attr('class', 'xx axis')
    .attr('transform', 'translate(0,' + height + ')')
    .call(xx.axis = d3.svg.axis().scale(xx).orient('bottom'))

var paths = svg2.append('g')

for (var name in groups) {
    var group = groups[name]
    group.path = paths.append('path')
        .data([group.data])
        .attr('class', name + ' group')
}

function tick() {

    now = new Date()

    d3.json("testdata.json", function (err, data){
        if (err) return console.warn(err);

        var newData = data;
        data.forEach(function(d,i){
            var newValue = d.value + Math.floor((Math.random()*10) - 5)
            newData[i].value = newValue <= 0 ? 10 : newValue
        });

        var obj = {
            'democrat': 0,
            'republican': 0
        };

        for (var j = 0; j < newData.length; j++) {
            obj[newData[j].key] = newData[j].value;
        }
    });
}

```

```
// Add new values
for (var name in groups) {
  var group = groups[name]
  //group.data.push(group.value) // Real values arrive at irregular intervals
  group.data.push(obj[name])
  group.path.style('stroke', group.color)
  group.path.attr('d', line)
}
```

```
// Shift domain
xx.domain([now - (limit - 2) * duration, now - duration])
```

```
// Slide xx-axis left
axis.transition()
  .duration(duration)
  .ease('linear')
  .call(xx.axis)
```

```
// Slide paths left
paths.attr('transform', null)
  .transition()
  .duration(duration)
  .ease('linear')
  .attr('transform', 'translate(' + xx(now - (limit - 1) * duration) + ')')
  .each('end', tick)
```

```
// Remove oldest data point from each group
for (var name in groups) {
  var group = groups[name]
  group.data.shift()
}
```

```
});
```

```
}
```

```
tick()
</script>
```

```
</body>
```