



Pykafka Integration with Python/Flask

Sno	Date	Modification	Author	Verified By
1	2019/07/31	Initial Document	Nishtha	SumitGoyal

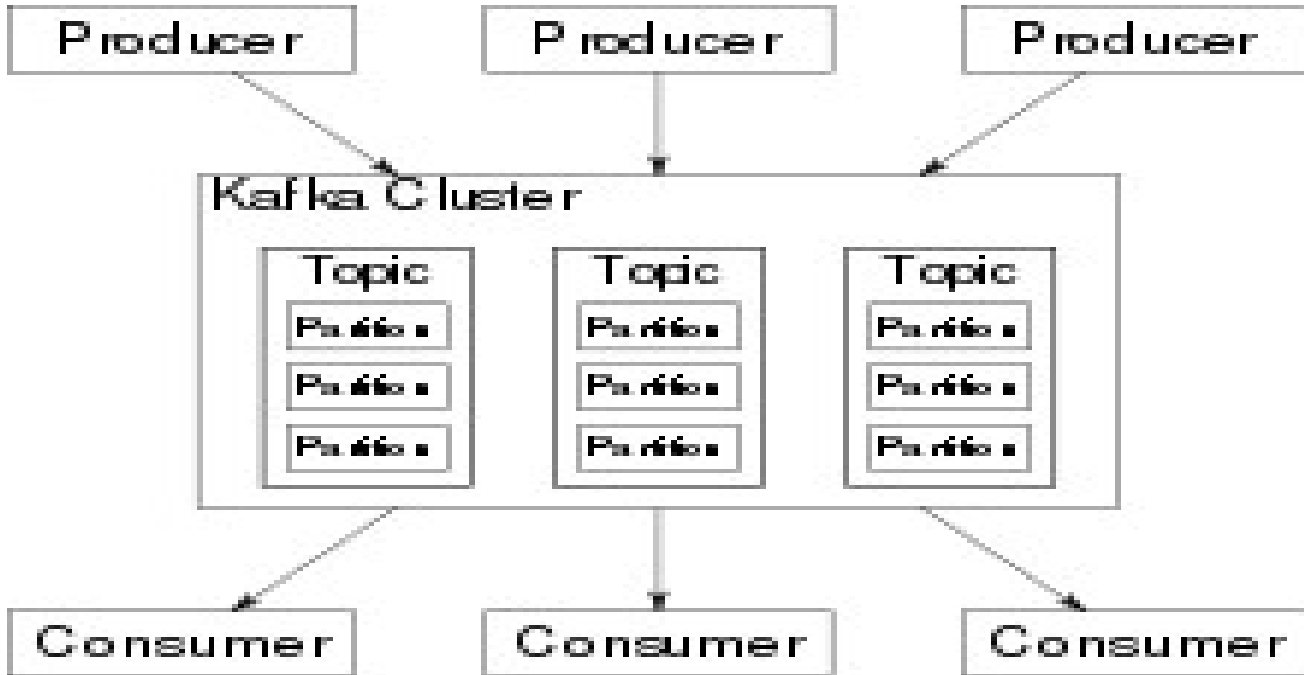
Table of Contents

Pykafka integration with python/flask	Error! Bookmark not defined.
Business Requirement.....	3
Solutions:.....	4
Install kafka on windows	Error! Bookmark not defined.
install java.....	Error! Bookmark not defined.
Steps to use python snowflake for Integration	Error! Bookmark not defined.
Code to save data into Snowflake cloud from insert data page.....	Error! Bookmark not defined.
Testing:	Error! Bookmark not defined.
Steps to test Integartion.....	Error! Bookmark not defined.
Final Result.	Error! Bookmark not defined.

Pykafka Integraion with Python/flask

PyKafka is a programmer-friendly Kafka client for Python. It includes Python implementations of Kafka producers and consumers, which are optionally backed by a C extension built on [librdkafka](#).

PyKafka's primary goal is to provide a similar level of abstraction to the [JVM Kafka client](#) using idioms familiar to Python programmers and exposing the most Pythonic API possible..



Business Requirement

The main objective of this project is that to build a live map of india with realtimeupdates..we will use apache kafka,javascript and python(flask Pykafk and json)

Solutions:

Note:In this document we explain step by step Integration between Python/flask and kafka (to show live map) using pykafka.

Steps :

Download java

We can download java from below URL-

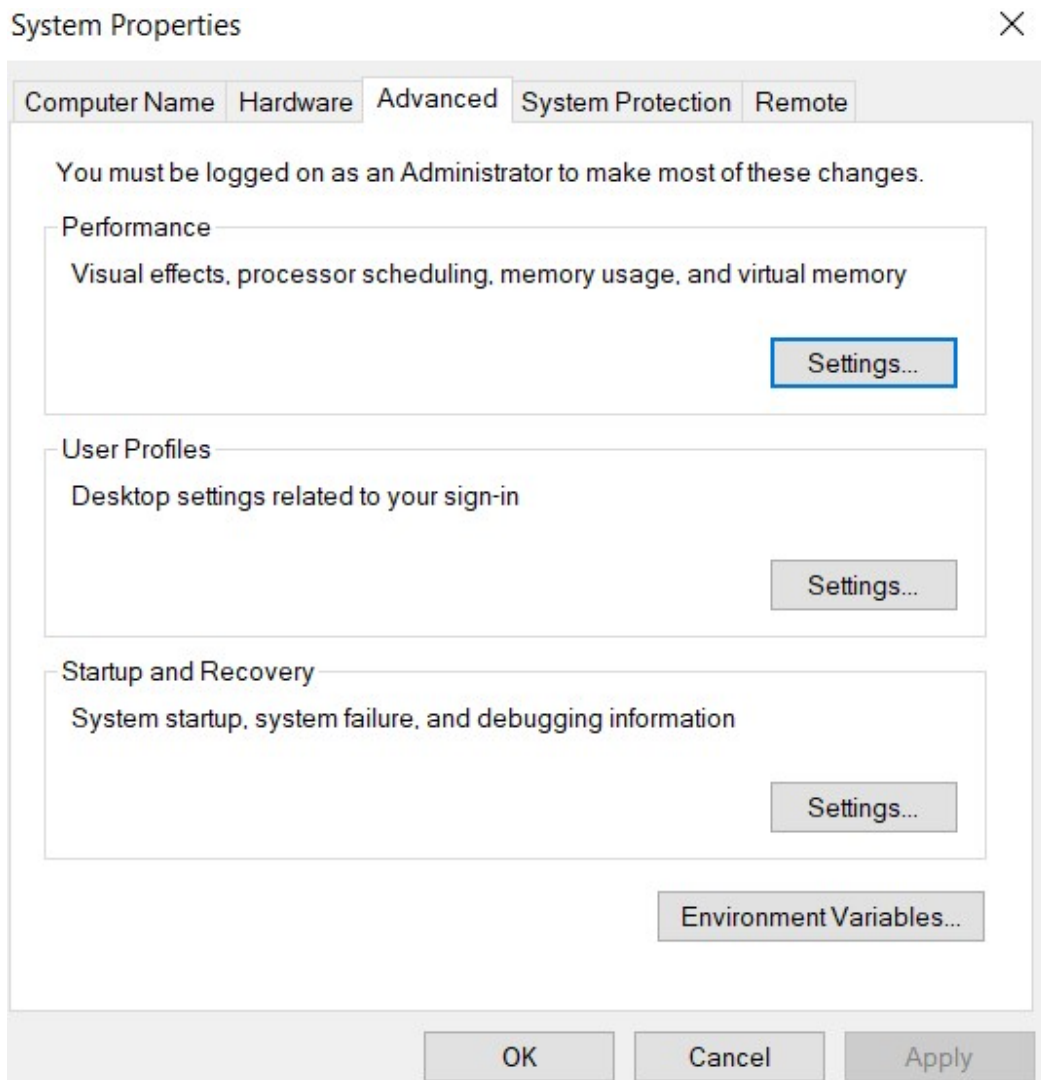
Link:<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Java SE Development Kit 8u221		
You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.		
Thank you for accepting the Oracle Technology Network License Agreement for Oracle Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.9 MB	jdk-8u221-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.81 MB	jdk-8u221-linux-arm64-vfp-hflt.tar.gz
Linux x86	174.18 MB	jdk-8u221-linux-i586.rpm
Linux x86	189.03 MB	jdk-8u221-linux-i586.tar.gz
Linux x64	171.19 MB	jdk-8u221-linux-x64.rpm
Linux x64	186.06 MB	jdk-8u221-linux-x64.tar.gz
Mac OS X x64	252.52 MB	jdk-8u221-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	132.99 MB	jdk-8u221-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.23 MB	jdk-8u221-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.66 MB	jdk-8u221-solaris-x64.tar.Z
Solaris x64	91.95 MB	jdk-8u221-solaris-x64.tar.gz
Windows x86	202.73 MB	jdk-8u221-windows-i586.exe
Windows x64	215.35 MB	jdk-8u221-windows-x64.exe

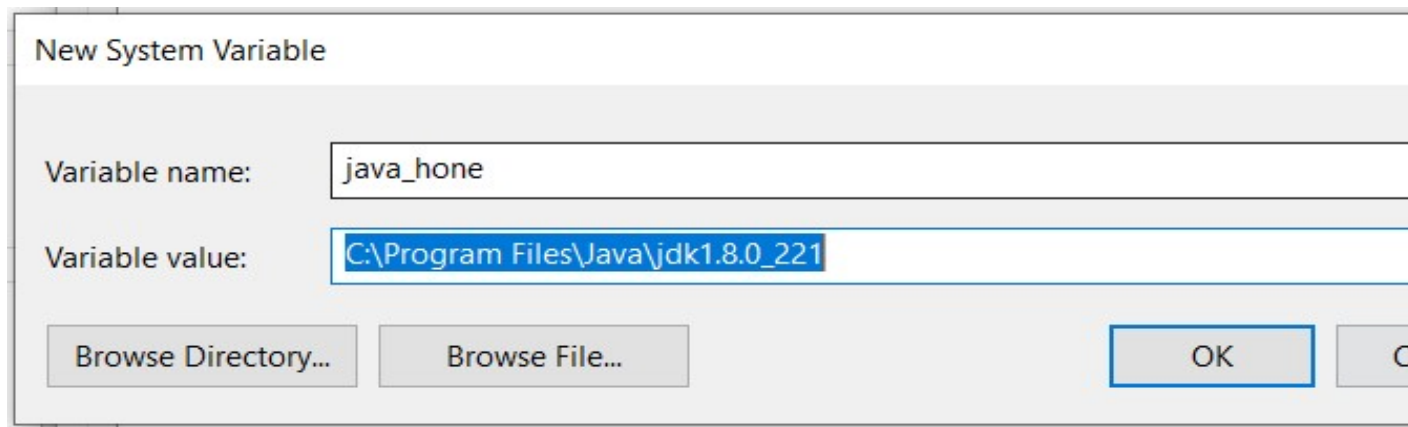
Once java is downloaded need to install it..and set the path in environment variable that is present in advance system settings.

- a) Click on advance system settings.

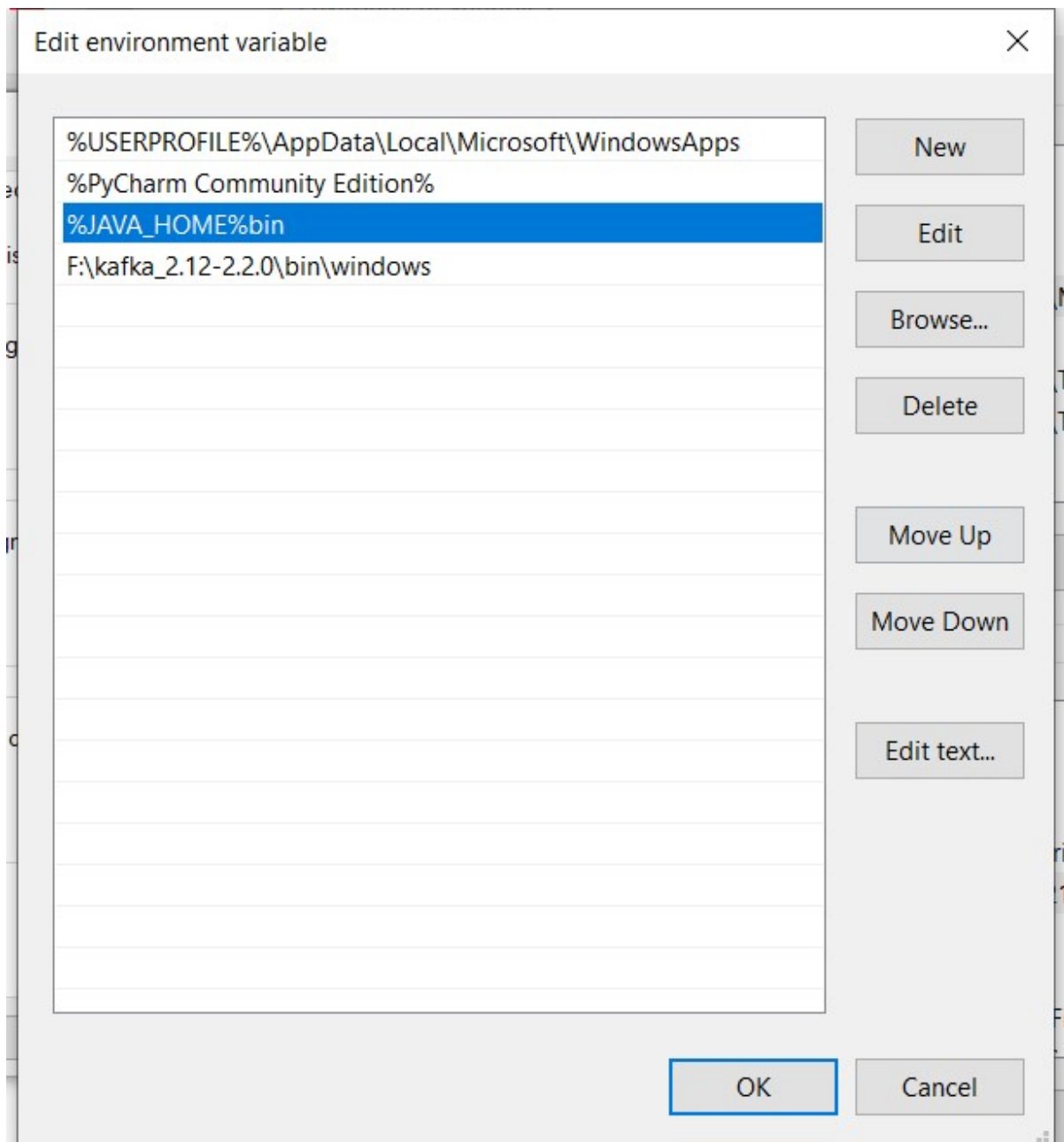
b) Click on environment variables.



Click on environment variable..then click on system variable and set the path of java



And in user variable click on path then edit button and set the java path.



Once the path is set now you can check the java version by enter the cmd

>> java - version in command prompt.

2) Download apache kafka by clicking on the below url.

https://www.apache.org/dyn/closer.cgi?path=/kafka/2.2.0/kafka_2.12-2.2.0.tgz

HTTP

http://apachemirror.wuchna.com/kafka/2.2.0/kafka_2.12-2.2.0.tgz

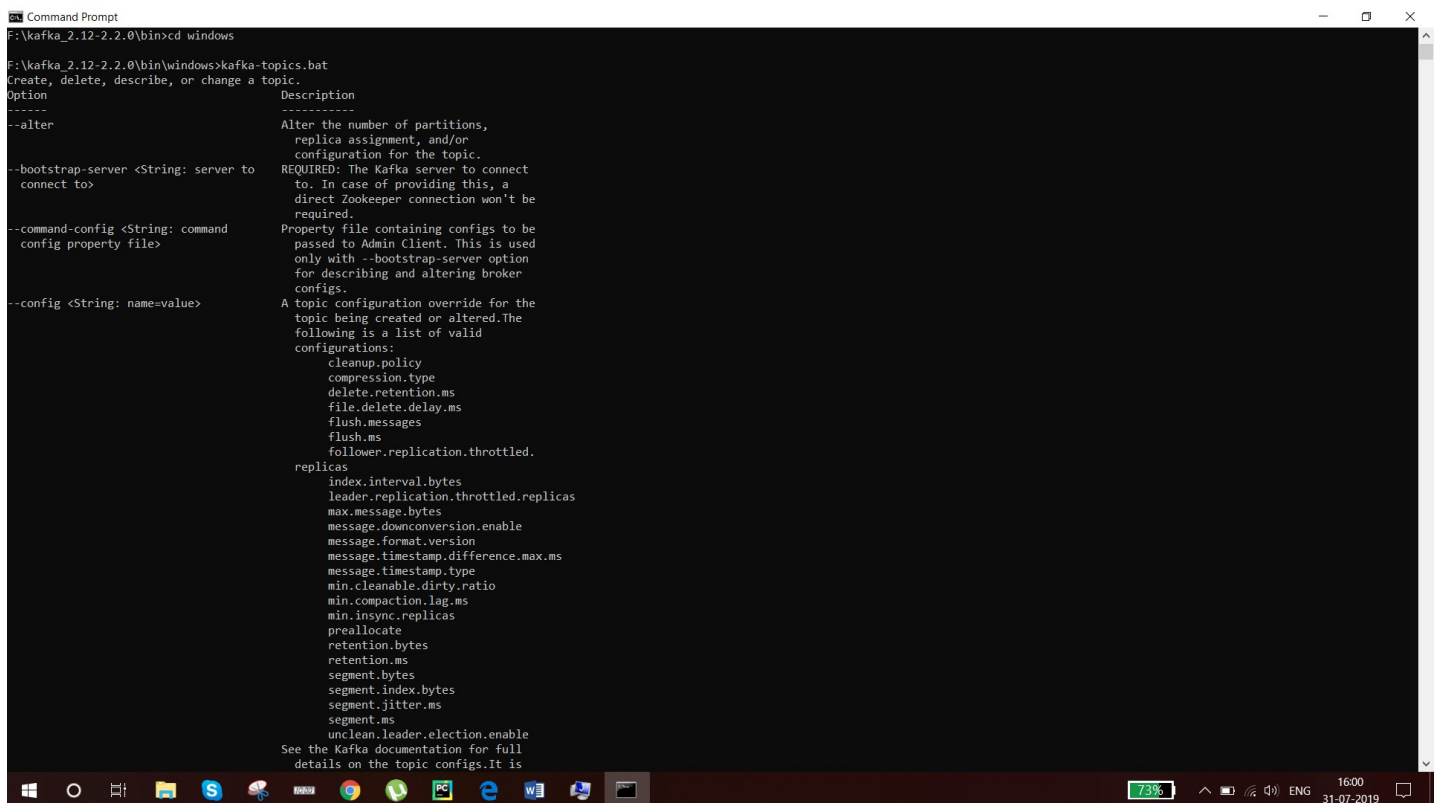
Once kafka has downloaded..unzip it

Open the commad prompt and go to the directory where you unzip kafka folder

And hit the cmd

```
>>f:\kafka_2.12-2.2.0\bin\winfows>kafka-topics.bat
```

Hit enter..ifou get something like this.

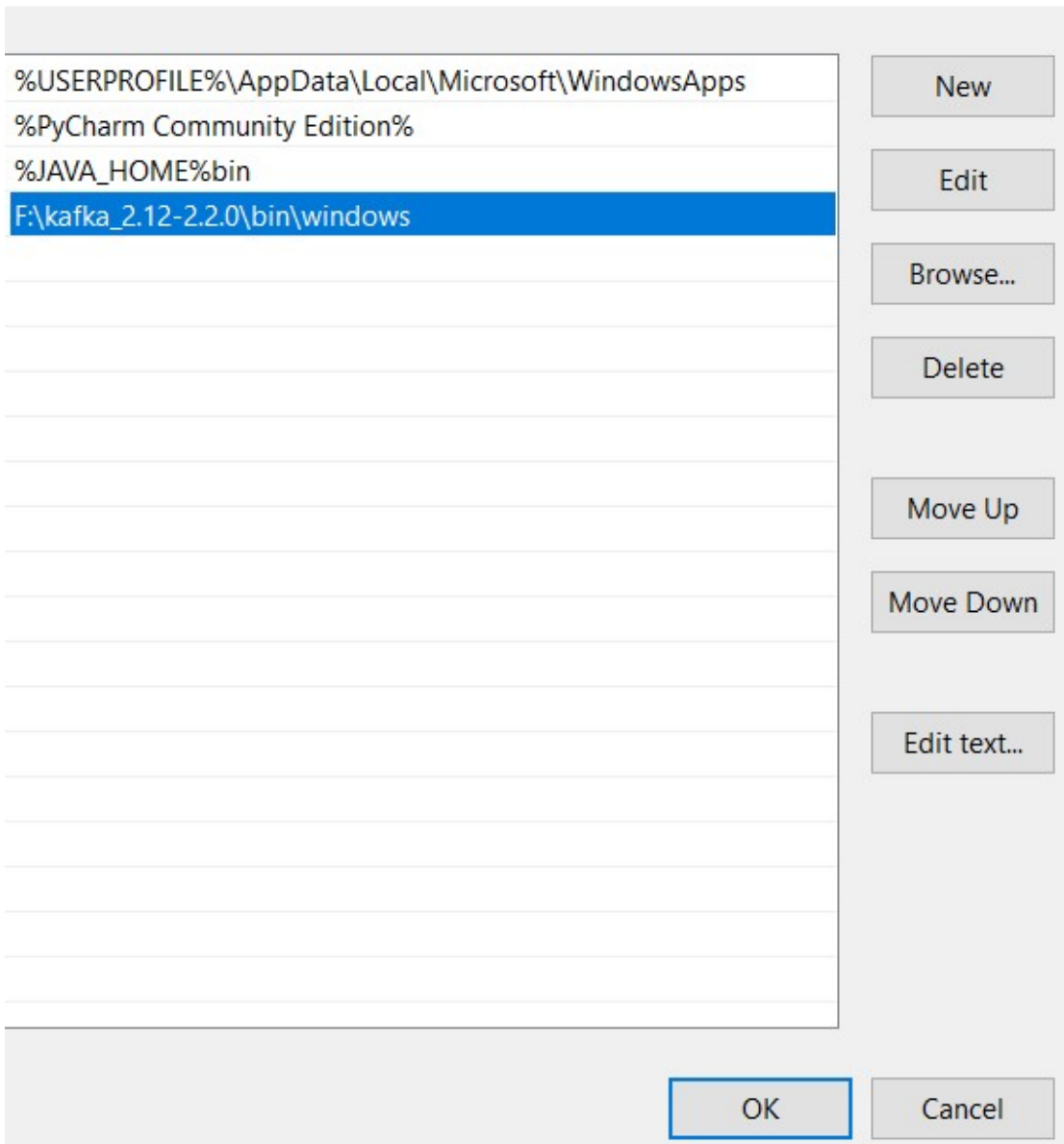


```
Command Prompt
F:\kafka_2.12-2.2.0\bin>cd windows
F:\kafka_2.12-2.2.0\bin\windows>kafka-topics.bat
Create, delete, describe, or change a topic.
Option      Description
-----
--alter      Alter the number of partitions,
             replica assignment, and/or
             configuration for the topic.
--bootstrap-server <String: server to
connect to>  REQUIRED: The Kafka server to connect
             to. In case of providing this, a
             direct Zookeeper connection won't be
             required.
--command-config <String: command
config property file> Property file containing configs to be
             passed to Admin Client. This is used
             only with --bootstrap-server option
             for describing and altering broker
             configs.
--config <String: name=value> A topic configuration override for the
             topic being created or altered.The
             following is a list of valid
             configurations:
             cleanup.policy
             compression.type
             delete.retention.ms
             file.delete.delay.ms
             flush.messages
             flush.ms
             follower.replication.throttled.
             replicas
             index.interval.bytes
             leader.replication.throttled.replicas
             max.message.bytes
             message.downconversion.enable
             message.format.version
             message.timestamp.difference.max.ms
             message.timestamp.type
             min.cleanable.dirty.ratio
             min.compaction.lag.ms
             min.insync.replicas
             preallocate
             retention.bytes
             retention.ms
             segment.bytes
             segment.index.bytes
             segment.jitter.ms
             segment.ms
             unclean.leader.election.enable
             See the Kafka documentation for full
             details on the topic configs.It is
```

Congrats you have successfully installed kafka in your windows.

We need to set the kafka path in enviroment variable for proper use of kafka server

lit environment variable



How to start kafka in windows:

Step1: Go to the directory where kafka is installed..

Step2: make a folder called data

Step3: under data again create two folder kafka and the another one is zookeeper..we need to this folder for storing logs

Step4: we need to modify the zookeeper path n zookeeper.py file

F:\kafka_2.12-2.2.0\config under this directory


```
dataDir=F:/kafka_2.12-2.2.0/data/zookeeper
# the port at which the clients will connect
clientPort=2181
# disable the per-ip limit on the number of connections since this is a non-production config
maxClientCnxns=0
```

Step5: need to modify the kafka server path

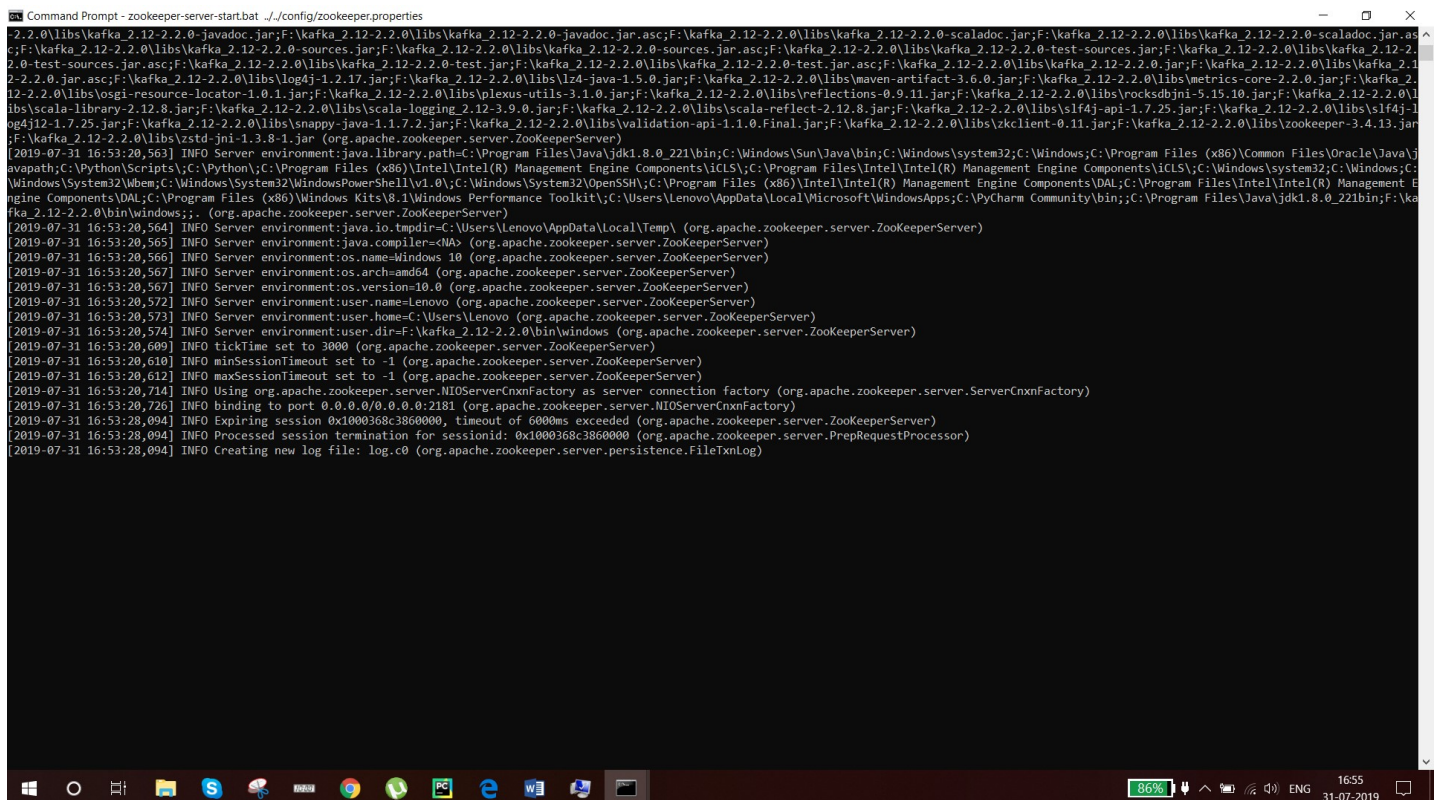
F:\kafka_2.12-2.2.0\config under server properties.

```
##### Log Basics #####
# A comma separated list of directories under which to store log files
log.dirs=F:/kafka_2.12-2.2.0/data/kafka
# The default number of log partitions per topic. More partitions allow greater
# parallelism for consumption, but this will also result in more files across
# the brokers.
num.partitions=1
zookeeper.connect=0.0.0.0:2181
```

Step6: need to start zookeeper server by entering the following cmd in cmd prompt

>>F:\kafka_2.12-2.2.0\bin\windows>zookeeper-server-start.bat ../../config/zookeeper.properties

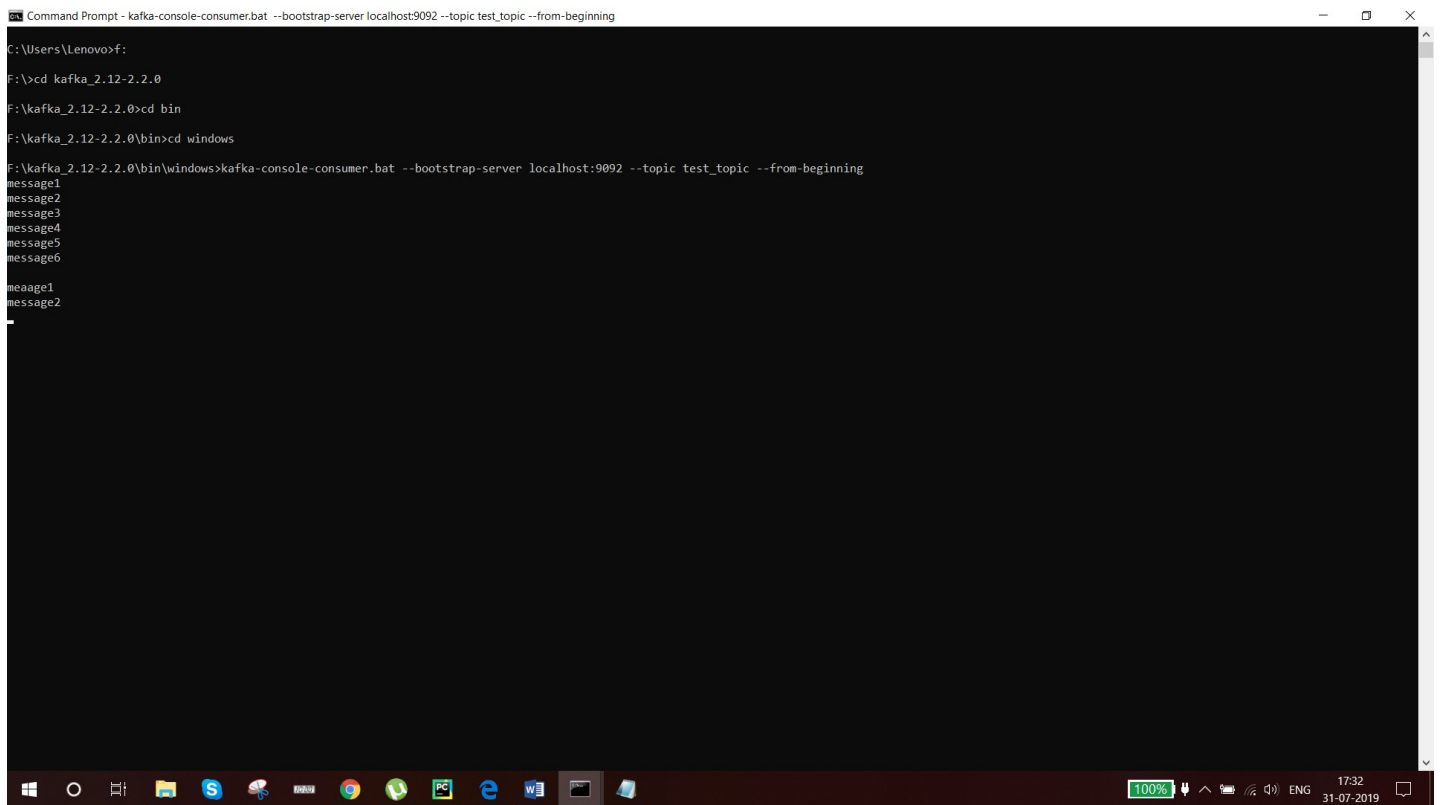
If you see the following screen then your zookeeper server is up to running



```
Command Prompt - zookeeper-server-start.bat ../../config/zookeeper.properties
-2.2.0\libs\kafka_2.12-2.2.0-javadoc.jar;F:\kafka_2.12-2.2.0\libs\kafka_2.12-2.2.0-javadoc.jar.asc;F:\kafka_2.12-2.2.0\libs\kafka_2.12-2.2.0-scaladoc.jar;F:\kafka_2.12-2.2.0\libs\kafka_2.12-2.2.0-scaladoc.jar.as
c;F:\kafka_2.12-2.2.0\libs\kafka_2.12-2.2.0-sources.jar;F:\kafka_2.12-2.2.0\libs\kafka_2.12-2.2.0-sources.jar.asc;F:\kafka_2.12-2.2.0\libs\kafka_2.12-2.2.0-test-sources.jar;F:\kafka_2.12-2.2.0\libs\kafka_2.12-2.
2.0-test-sources.jar.asc;F:\kafka_2.12-2.2.0\libs\kafka_2.12-2.2.0-test.jar;F:\kafka_2.12-2.2.0\libs\kafka_2.12-2.2.0-test.jar.asc;F:\kafka_2.12-2.2.0\libs\kafka_2.12-2.2.0.jar;F:\kafka_2.12-2.2.0\libs\kafka_2.12-2.2.0.jar.asc;F:\kafka_2.12-2.2.0\libs\log4j-1.2.17.jar;F:\kafka_2.12-2.2.0\libs\log4j-1.2.17.jar.asc;F:\kafka_2.12-2.2.0\libs\log4j-core-1.2.17.jar;F:\kafka_2.12-2.2.0\libs\log4j-core-1.2.17.jar.asc;F:\kafka_2.12-2.2.0\libs\plexus-utils-3.1.0.jar;F:\kafka_2.12-2.2.0\libs\plexus-utils-3.1.0.jar.asc;F:\kafka_2.12-2.2.0\libs\reflections-0.9.11.jar;F:\kafka_2.12-2.2.0\libs\reflections-0.9.11.jar.asc;F:\kafka_2.12-2.2.0\libs\rocksdbjni-5.15.10.jar;F:\kafka_2.12-2.2.0\l
ibs\scala-library-2.12.8.jar;F:\kafka_2.12-2.2.0\libs\scala-logging_2.12-3.9.0.jar;F:\kafka_2.12-2.2.0\libs\scala-reflect-2.12.8.jar;F:\kafka_2.12-2.2.0\libs\slf4j-api-1.7.25.jar;F:\kafka_2.12-2.2.0\libs\slf4j-l
og4j12-1.7.25.jar;F:\kafka_2.12-2.2.0\libs\snappy-java-1.1.7.2.jar;F:\kafka_2.12-2.2.0\libs\validation-api-1.1.0.Final.jar;F:\kafka_2.12-2.2.0\libs\zkclient-0.11.jar;F:\kafka_2.12-2.2.0\libs\zookeeper-3.4.13.jar
;F:\kafka_2.12-2.2.0\libs\zstd-jni-1.3.8-1.jar (org.apache.zookeeper.server.ZooKeeperServer)
[2019-07-31 16:53:20,563] INFO Server environment:java.library.path=C:\Program Files\Java\jdk1.8.0_221\bin;C:\Windows\Sun\Java\bin;C:\Windows\system32;C:\Windows;C:\Windows\System32\WindowsPowerShell\v1.0;C:\Windows\System32\OpenSSH;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files\Intel\Intel(R) Management E
ngine Components\DAL;C:\Program Files (x86)\Windows Kits\8.1\Windows Performance Toolkit;C:\Users\Lenovo\AppData\Local\Microsoft\WindowsApps;C:\PyCharm Community\bin;C:\Program Files\Java\jdk1.8.0_221\bin;F:\ka
fka_2.12-2.2.0\bin\windows;. (org.apache.zookeeper.server.ZooKeeperServer)
[2019-07-31 16:53:20,564] INFO Server environment:java.io.tmpdir=C:\Users\Lenovo\AppData\Local\Temp (org.apache.zookeeper.server.ZooKeeperServer)
[2019-07-31 16:53:20,565] INFO Server environment:java.compiler=cNA> (org.apache.zookeeper.server.ZooKeeperServer)
[2019-07-31 16:53:20,566] INFO Server environment:os.name=Windows 10 (org.apache.zookeeper.server.ZooKeeperServer)
[2019-07-31 16:53:20,567] INFO Server environment:os.arch=amd64 (org.apache.zookeeper.server.ZooKeeperServer)
[2019-07-31 16:53:20,567] INFO Server environment:os.version=10.0 (org.apache.zookeeper.server.ZooKeeperServer)
[2019-07-31 16:53:20,572] INFO Server environment:user.name=Lenovo (org.apache.zookeeper.server.ZooKeeperServer)
[2019-07-31 16:53:20,573] INFO Server environment:user.home=C:\Users\Lenovo (org.apache.zookeeper.server.ZooKeeperServer)
[2019-07-31 16:53:20,574] INFO Server environment:user.dir=F:\kafka_2.12-2.2.0\bin\windows (org.apache.zookeeper.server.ZooKeeperServer)
[2019-07-31 16:53:20,609] INFO TickTime set to 3000 (org.apache.zookeeper.server.ZooKeeperServer)
[2019-07-31 16:53:20,610] INFO minSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2019-07-31 16:53:20,612] INFO maxSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2019-07-31 16:53:20,714] INFO Using org.apache.zookeeper.server.NIOServerCnxnFactory as server connection factory (org.apache.zookeeper.server.ServerCnxnFactory)
[2019-07-31 16:53:20,726] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2019-07-31 16:53:28,094] INFO Expiring session 0x1000368c3860000, timeout of 6000ms exceeded (org.apache.zookeeper.server.ZooKeeperServer)
[2019-07-31 16:53:28,094] INFO Processed session termination for sessionId: 0x1000368c3860000 (org.apache.zookeeper.server.PreRequestProcessor)
[2019-07-31 16:53:28,094] INFO Creating new log file: log.c0 (org.apache.zookeeper.server.persistence.FileTxnLog)
```



```
>>F:\kafka_2.12-2.2.0\bin\windows>kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test_topic --from-beginning
```



```
Command Prompt - kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test_topic --from-beginning
C:\Users\Lenovo>F:
F:\>cd kafka_2.12-2.2.0
F:\kafka_2.12-2.2.0>cd bin
F:\kafka_2.12-2.2.0\bin>cd windows
F:\kafka_2.12-2.2.0\bin\windows>kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test_topic --from-beginning
message1
message2
message3
message4
message5
message6

message1
message2
```

Connect to flask:

```
>>pip install pykafka
```

Here we are generating live bus location

So we need to generate a map api by hit the below url

<https://account.mapbox.com/>

once you successfully logged in. you see access token in your map box dashboard copy it

busdata1.py:

```
from pykafka import KafkaClient
import json
from datetime import datetime
import uuid
import time
```

```

#READ COORDINATES FROM GEOJSON
input_file = open('bus1.json')
json_array = json.load(input_file)
coordinates = json_array['features'][0]['geometry']['coordinates']

#GENERATE UUID
defgenerate_uuid():
return uuid.uuid4()

#KAFKA PRODUCER
client = KafkaClient(hosts="localhost:9092")
topic = client.topics['geodata_final123']
producer = topic.get_sync_producer()

#CONSTRUCT MESSAGE AND SEND IT TO KAFKA
data = {}
data['busline'] = '00001'

defgenerate_checkpoint(coordinates):
i = 0
while i<len(coordinates):
    data['key'] = data['busline'] + '_' + str(generate_uuid())
    data['timestamp'] = str(datetime.utcnow())
    data['latitude'] = coordinates[i][1]
    data['longititude'] = coordinates[i][0]
    message = json.dumps(data)
print(message)
producer.produce(message.encode('ascii'))
time.sleep(1)

#if bus reaches last coordinate, start from beginning
if i == len(coordinates)-1:
i = 0
else:
i += 1

generate_checkpoint(coordinates)

```

index.html:

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<!-- LEAFLET -->
<link rel="stylesheet" href="https://unpkg.com/leaflet@1.4.0/dist/leaflet.css"
integrity="sha512-
puBpdR07980ZvTTbP4A8Ix/1+A4dHDD0DGqYW6RQ+9jxkRFclaxxQb/SJAWZfWakuyeQUyt07+7N4QKrDh+drA=="
crossorigin=""/>
<script src="https://unpkg.com/leaflet@1.4.0/dist/leaflet.js"
integrity="sha512-
QVftwZFqvtRNi0ZyCtsznlKSW0StnDORoeFr1enyq5mVL4t4mKB3S/EnC3rRjcxCPavG10IcrVGSmPh6Qw5lwng=="
crossorigin=""></script>
<!-- END LEAFLET -->
<title>London Live Map</title>
</head>
<body>
<h1>London Bus Live Map</h1>

```

```

<!-- LEAFLET -->
<div id="mapid" style = "width:900px; height:580px;"></div>
<script src="../static/leaflet.js"></script>
<!-- END LEAFLET -->
</body>
</html>

```

Leaf.js:

```

var mymap = L.map('mapid').setView([51.505, -0.09], 13);
L.tileLayer('https://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}.png?access_token={accessToken}', {
  attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a>
  contributors, <a href="https://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, Imagery © <a
  href="https://www.mapbox.com/">Mapbox</a>',
  maxZoom: 18,
  id: 'mapbox.streets',
  accessToken: 'pk.eyJ1IjoibmlzaHRoYTAzIiw6ImNqZXBuZGxtZDhhdXczbm9mMHkyMHc4cGEifQ.LM0dABNaOv2m-
  phFTKjtUQ' //ENTER YOUR ACCESS TOKEN HERE
}).addTo(mymap);

mapMarkers1 = [];
mapMarkers2 = [];
mapMarkers3 = [];

var source = new EventSource('/topic/TOPICNAME'); //ENTER YOUR TOPICNAME HERE
source.addEventListener('message', function(e){

  console.log('Message');
  obj = JSON.parse(e.data);
  console.log(obj);

  if(obj.busline == '00001') {
    for (var i = 0; i < mapMarkers1.length; i++) {
      mymap.removeLayer(mapMarkers1[i]);
    }
    marker1 = L.marker([obj.latitude, obj.longitude]).addTo(mymap);
    mapMarkers1.push(marker1);
  }

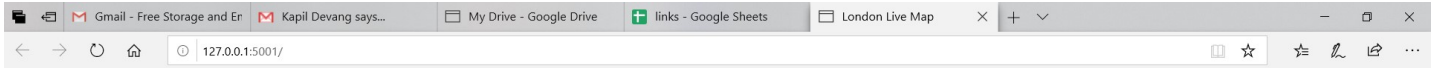
  if(obj.busline == '00002') {
    for (var i = 0; i < mapMarkers2.length; i++) {
      mymap.removeLayer(mapMarkers2[i]);
    }
    marker2 = L.marker([obj.latitude, obj.longitude]).addTo(mymap);
    mapMarkers2.push(marker2);
  }

  if(obj.busline == '00003') {
    for (var i = 0; i < mapMarkers3.length; i++) {
      mymap.removeLayer(mapMarkers3[i]);
    }
    marker3 = L.marker([obj.latitude, obj.longitude]).addTo(mymap);
    mapMarkers3.push(marker3);
  }
}, false);

```

final output:

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help kafka_busdata [F:\kafka_busdata] - |busdata1.py - PyCharm
Project kafka_busdata busdata1.py busdata2.py busdata3.py bus2.json bus3.json app.py index.html leaf.js bus1.json
Run app x busdata1 x
F:\kafka_busdata\venv\Scripts\python.exe F:/kafka_busdata/busdata1.py
{"busline": "00001", "key": "00001_e72e904-32ef-48a7-a6ec-4846ed163fc", "timestamp": "2019-07-31 12:21:34.504936", "latitude": 51.51093265116127, "longitude": -0.10479927862988281}
{"busline": "00001", "key": "00001_deaf2440-4a44-44d0-afe9-2ddf153dd45b", "timestamp": "2019-07-31 12:21:35.543409", "latitude": 51.510077954475555, "longitude": -0.1185321807861328}
{"busline": "00001", "key": "00001_c9e55c9-50e7-498f-9706-16676b71010b", "timestamp": "2019-07-31 12:21:36.549348", "latitude": 51.50478916217527, "longitude": -0.11342525482177734}
{"busline": "00001", "key": "00001_2eebe416-ef92-4f3f-e4b0-c1b5c2bfeae8", "timestamp": "2019-07-31 12:21:37.549755", "latitude": 51.50713981232172, "longitude": -0.1078033447265625}
{"busline": "00001", "key": "00001_dc39addf-d393-411b-bcdc-fc8ba8d943d3", "timestamp": "2019-07-31 12:21:38.559391", "latitude": 51.50700625590363, "longitude": -0.10342597961425781}
{"busline": "00001", "key": "00001_e3588eba-4505-4f3a-9fd9-59d6f5a50e7e", "timestamp": "2019-07-31 12:21:39.563588", "latitude": 51.511173031715074, "longitude": -0.10342597961425781}
{"busline": "00001", "key": "00001_65d92e80-19ce-4db0-a904-2a46651e0cc2", "timestamp": "2019-07-31 12:21:40.564092", "latitude": 51.51098606917176, "longitude": -0.10454177856445312}
{"busline": "00001", "key": "00001_e3fd3fef-2c5e-4c33-878a-ef7e537e1736", "timestamp": "2019-07-31 12:21:41.573961", "latitude": 51.509223241755116, "longitude": -0.1059579849243164}
{"busline": "00001", "key": "00001_4d160a1e-191b-44f7-ab00-67a12839c4cb", "timestamp": "2019-07-31 12:21:42.579055", "latitude": 51.51050530482274, "longitude": -0.10544300079345703}
{"busline": "00001", "key": "00001_17c7cf2-c151-4ea3-ab11-71a55e7dce45", "timestamp": "2019-07-31 12:21:43.579658", "latitude": 51.51093265116127, "longitude": -0.10479927862988281}
{"busline": "00001", "key": "00001_2a9bd865-8d7a-4feb-b97c-9d30eb3a893c", "timestamp": "2019-07-31 12:21:44.581501", "latitude": 51.510077954475555, "longitude": -0.1185321807861328}
{"busline": "00001", "key": "00001_986f3bc4-2a8a-4f0c-9007-6ad3f7287479", "timestamp": "2019-07-31 12:21:45.597870", "latitude": 51.50478916217527, "longitude": -0.11342525482177734}
{"busline": "00001", "key": "00001_1a857500-6c11-435c-aaa1-8d825f4c4d93", "timestamp": "2019-07-31 12:21:46.604400", "latitude": 51.50713981232172, "longitude": -0.1078033447265625}
{"busline": "00001", "key": "00001_c2acd0b1-ac03-4b78-b3b0-2e4017432f2a", "timestamp": "2019-07-31 12:21:47.605339", "latitude": 51.50700625590363, "longitude": -0.10342597961425781}
{"busline": "00001", "key": "00001_f603c71c-2a9b-40b8-b44f-4440a4d36d4b", "timestamp": "2019-07-31 12:21:48.608295", "latitude": 51.511173031715074, "longitude": -0.10342597961425781}
{"busline": "00001", "key": "00001_28f9d9ae-145a-48ac-a401-f74c76da1dfd", "timestamp": "2019-07-31 12:21:49.618456", "latitude": 51.51098606917176, "longitude": -0.10454177856445312}
{"busline": "00001", "key": "00001_6366d392-a262-4f18-b4ac-e2a84d71eb44", "timestamp": "2019-07-31 12:21:50.624084", "latitude": 51.509223241755116, "longitude": -0.1059579849243164}
{"busline": "00001", "key": "00001_8a921579-e73d-444b-ac87-c193fb277699", "timestamp": "2019-07-31 12:21:51.625083", "latitude": 51.51050530482274, "longitude": -0.10544300079345703}
{"busline": "00001", "key": "00001_100b3684-56bc-475c-80bb-207e78c502b6", "timestamp": "2019-07-31 12:21:52.633004", "latitude": 51.51093265116127, "longitude": -0.10479927862988281}
{"busline": "00001", "key": "00001_6391d0a7-ab90-440d-86a9-3756baa6704f", "timestamp": "2019-07-31 12:21:53.634919", "latitude": 51.510077954475555, "longitude": -0.1185321807861328}
{"busline": "00001", "key": "00001_88f9d536-588c-4054-a705-f1c28b1db675", "timestamp": "2019-07-31 12:21:54.645857", "latitude": 51.50478916217527, "longitude": -0.11342525482177734}
{"busline": "00001", "key": "00001_3af40392-841d-4f8a-94fe-5555e00564ca", "timestamp": "2019-07-31 12:21:55.650089", "latitude": 51.50713981232172, "longitude": -0.1078033447265625}
{"busline": "00001", "key": "00001_e3e0cd73-881d-4d8e-8d78-e9832bb99536", "timestamp": "2019-07-31 12:21:56.650839", "latitude": 51.50700625590363, "longitude": -0.10342597961425781}
{"busline": "00001", "key": "00001_f6e6112a-80f1-425c-8df3-81a10118b92a", "timestamp": "2019-07-31 12:21:57.653584", "latitude": 51.511173031715074, "longitude": -0.10342597961425781}
{"busline": "00001", "key": "00001_49d3a349-ab87-4d41-b4c5-bb0ed924bb1d", "timestamp": "2019-07-31 12:21:58.664497", "latitude": 51.51098606917176, "longitude": -0.10454177856445312}
{"busline": "00001", "key": "00001_0a7d3e38-c296-4bbe-9e05-32e1f01e72cb", "timestamp": "2019-07-31 12:21:59.667762", "latitude": 51.509223241755116, "longitude": -0.1059579849243164}
{"busline": "00001", "key": "00001_8a9549b7-4c4e-40cf-930f-74aa129a6e77", "timestamp": "2019-07-31 12:22:00.668172", "latitude": 51.51050530482274, "longitude": -0.10544300079345703}
{"busline": "00001", "key": "00001_80185d9e-17af-4eb0-8b4b-e8450574ac18", "timestamp": "2019-07-31 12:22:01.670970", "latitude": 51.51093265116127, "longitude": -0.10479927862988281}
{"busline": "00001", "key": "00001_5a899e12-337e-460c-8e20-56dad453ea1f", "timestamp": "2019-07-31 12:22:02.676028", "latitude": 51.510077954475555, "longitude": -0.1185321807861328}
{"busline": "00001", "key": "00001_cb4d2e51-3326-497a-b6c4-285b0aaafc70", "timestamp": "2019-07-31 12:22:03.685011", "latitude": 51.50478916217527, "longitude": -0.11342525482177734}
{"busline": "00001", "key": "00001_6ad3b08b-b67f-4476-8e2e-132e85d4895f", "timestamp": "2019-07-31 12:22:04.689143", "latitude": 51.50713981232172, "longitude": -0.1078033447265625}
{"busline": "00001", "key": "00001_8999d474-2c33-41c1-a59a-0b57ebf409f2", "timestamp": "2019-07-31 12:22:05.689906", "latitude": 51.50700625590363, "longitude": -0.10342597961425781}
Windows Defender might be impacting your build performance. PyCharm checked the following directories: //F:\kafka_busdata // C:\Users\Lenovo\PyCharmCE2019.2\sys... (today 12:24) 1899 chars, 11 line breaks 1.2 CRLF UTF-8 4 spaces Python 3.7 (kafka_busdata)
Event Log
```



London Bus Live Map

