Importing the Packages

In [9]:

```
import requests
import json
import time
import folium
from pyspark import SparkContext
from pyspark.streaming import StreamingContext
from IPython.display import HTML
from threading import Thread
from IPython.display import IFrame
```

Creating a centered map at co-ordinates (0,0)

In [2]:

```
display_map = folium.Map(location=[0, 0], zoom_start=2, zoom_control=False, scrollWheelZoom=False)
```

We need to store the lattitude and longitude co-ordinates,so creating a list for them

In [3]:

```
coordinates = []
```

Setting up the url for the producer

In [4]:

```
url = "http://api.open-notify.org/iss-now.json"
```

Defining time interval and run time

In [5]:

```
inter = 5
run_time = 3600
start_time = time.time()
```

Defining a consumer which consumes data from producer

In [6]:

```
def consumer():
    while (time.time() - start_time) < run_time:
        try:
            response = requests.get(url)
            data = json.loads(response.content)

            longitude = float(data['iss_position']['longitude'])
            latitude = float(data['iss_position']['latitude'])

            folium.Marker([latitude, longitude]).add_to(display_map)

            coordinates.append((latitude, longitude))

            if len(coordinates) > 1:
                folium.PolyLine(coordinates, color="red", weight=5, opacity=1).add_to(display_map)

            time.sleep(inter)

        except Exception as e:
            print(e)
            continue
```

Defining a Producer

In [7]:

```
def producer():
    while (time.time() - start_time) < run_time:
        pass
```

Starting Spark Streaming with interval of 5sec

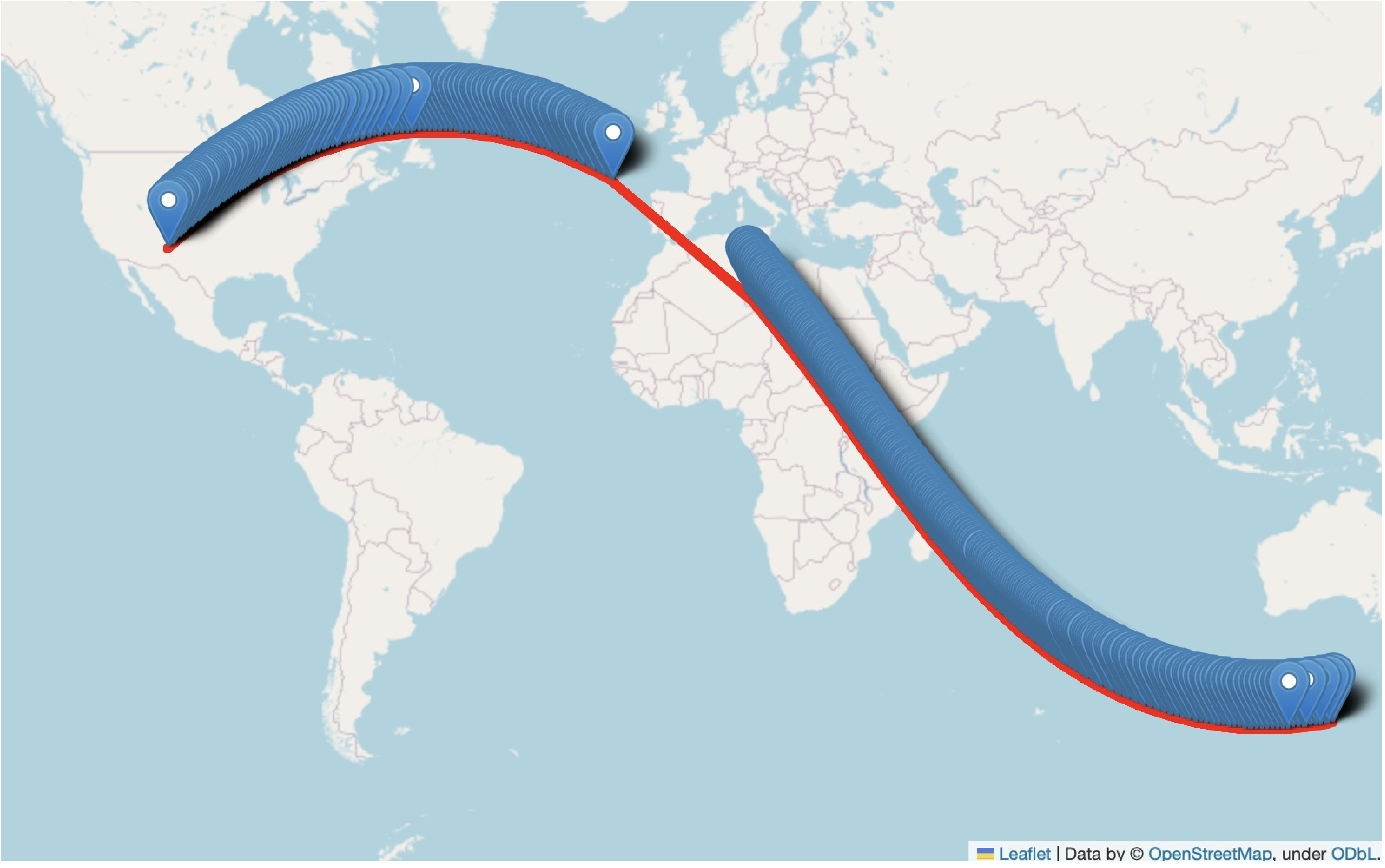
In [8]:

```
#Starting Spark Streaming
sc = SparkContext(appName="SatelliteTracking")
ssc = StreamingContext(sc, inter)
ssc.checkpoint("checkpoint")

#Spark Streaming Threading
Spark_stream_thread = Thread(target=consumer)
Spark_stream_thread.start()
producer()
Spark_stream_thread.join()

# Displaying the map in a Jupyter notebook
display_map.save('satellite_tracking.html')
IFrame(src='satellite_tracking.html', width=800, height=500)
```

Out[8]:



Interpretation: For completetion of this assignment, I have used various packages like requests,json,thread,spark streaming. Created a consumer which consumes the data from producer The map is plotted using folium and the satellite is tracked after starting the spark session and the tracking is done after every 5 seconds and then the co-ordinates are plotted and the line is drawn joining those co-ordinates of lattitude and longitute.

References

1. https://javadoc.io/doc/com.sparkjava/spark-core/2.6.0/spark/Request.html

1. https://www.json.org/json-en.html

1. https://python-visualization.github.io/folium/

1. https://www.databricks.com/resources/ebook/the-data-engineers-guide-to-apache-spark-and-delta-lake?utm_medium=paid+search&utm_source=google&utm_campaign=15029794441&utm_adgroup=141427216012&utm_content=ebook&utm_offer=the-data-engineers-guide-to-apache-spark-and-delta-lake&utm_ad=643092508759&utm_term=spark%20python&gclid=CjwKCAjwitShBhA6EiwAq3RqAwjQIxuIMaSimiWHBNC6RnCTm58BUMHmQ3iRIhBTC-R5llImFLtf6RoCZqgQAvD_BwE

1. https://www.databricks.com/resources/ebook/the-data-engineers-guide-to-apache-spark-and-delta-lake?utm_medium=paid+search&utm_source=google&utm_campaign=15029794441&utm_adgroup=141427216012&utm_content=ebook&utm_offer=the-data-engineers-guide-to-apache-spark-and-delta-lake&utm_ad=643092508759&utm_term=spark%20python&gclid=CjwKCAjwitShBhA6EiwAq3RqAwjQIxuIMaSimiWHBNC6RnCTm58BUMHmQ3iRIhBTC-R5llImFLtf6RoCZqgQAvD_BwE

1. https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.streaming.StreamingContext.html

2. https://docs.python.org/3/library/threading.html

3. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/iframe