Your solution must be able to handle any geojson file. The following tests are run for the evaluation:

1. Test: Start Observer application observe.py.

   ```
   python observe.py  <token> demo/#
   ```

Expected result: Success and ready to receive MQTT messages

```
Successfully connected to 'mqtt.flespi.io'
Successfully subscribed to topic 'demo/#'
```

2. Test case: Start simulator.

   ```
   python simulator.py <token> ./data/olten-brugg.geojson demo/frodo
   ```

   Expected result: Success and corresponding route is available on the serve

   ```
   Waiting to connect ...
   Connected with result code Success
   ```

   and file olten-brugg.kml is created after the simulation.

3. Test case: Start simulator with new data file, e.g. demo.geojson. Please note that this demo.geojson file is not included in the project. You can use any other file for your own tests. The tests should show that your Observer application does not have to be restarted to observe and correctly record a new route.Important: The Observer application from test case 1 is still active and will not be restarted.

   ```
   python simulator.py <token> ./data/demo.geojson demo/gimli
   ```

   Expected result: Success and corresponding route is available on the server

   ```
   Waiting to connect ...
   Connected with result code Success
   ```

   and file demo.kml is created after the simulation.

4. Case Test: Visualize route from file olten-brugg.kml in the KML viewer.