PROG 2700 ASSIGNMENT 3C – FLIGHT Tracker

**Client-Side Programming**

# Prerequisites

### Strongly Recommended: Assignment 3A & 3B

# Summary

You will create a solution that will do the following

* Display a map of the world. (You’ll be given starter files for this)
* Fetch real-time flight information data from a publicly available API.
* Filter the raw data to a subset with specific criteria. (plane itself(country of origin))
* Convert the filtered API data into GeoJSON format.( https://stackoverflow.com/questions/42523234/raw-data-to-array-or-json-using-js)
* Plot markers on the map to display the current position of a subset of aircraft.
* Add functionality that will cause the map to auto refresh after a certain interval of time.

# Display a map of the world.

For this assignment you will be working with the [**Leaflet.js**](https://leafletjs.com/) mapping library. Leaflet is a leading open-source JavaScript library for mobile-friendly interactive maps. It provides an easy-to-use programming API for customizing and building various types of maps.

Valuable Resources:

<https://leafletjs.com/reference-1.4.0.html>

<https://maptimeboston.github.io/leaflet-intro/> (Read this to get an intro to how Leaflet works!)

# Fetch real-time flight data

## API

The real-time flight data that you will leverage for this assignment can be accessed at [OpenSky Netowrk.](https://opensky-network.org/) The OpenSky Network is a community-based receiver network which continuously collects air traffic surveillance data. Unlike other networks, OpenSky keeps the collected raw data forever and makes it accessible to researchers. Real-time flight data can be accessed at <https://opensky-network.org/api/states/all>. This will return flight data for thousands of aircraft. Your application will need to fetch this data in its raw form and be able to filter the results according to the following criteria.

* **Requirement: Filter the resulting data so that you keep only those aircraft (states) whose country of origin is Canada.**

Refer to the OpenSky Network documentation to get an explanation of what data is returned and what format it presents itself.

## GeoJSON

Leaflet supports and works well with the [GeoJSON](http://geojson.org/) data format. It is a format for encoding a variety of geographic data structures and is widely used in the digital cartography industry.

You are required to transform your raw API data into GeoJSON format so that they can be applied to the map for point marking. Focus on Feature arrays or Feature Collections (either will work) when building out your formatted data.

Valuable Resources:

<https://macwright.org/2015/03/23/geojson-second-bite.html>

# Plot Markers on the map using the GeoJSON data.

Once you have your newly transformed data in GeoJSON format. Apply this data to the provided map using the programming API for GeoJSON in Leaflet.

Valuable Resources:

<https://leafletjs.com/examples/geojson/>

# Apply code to auto refresh the map.

Apply the following functionality to your app which will resemble how real-time flight tracking software behaves.

* After a certain period of time re-fetch the updated API data and perform the transformation as necessary.
* Refresh the map by re-rendering the markers in their new positions.

Note: adding JavaScript to cause the entire browser page to reload is not an acceptable solution for this requirement.

Note: be careful with this requirement. You should never trigger a re-fetch of your data until the previous fetch has been completed and processed. Otherwise, you may cause unintended results in your application.

# Other requirements

The following requirements are considered less critical to your application but will add to your overall mark for the assignment.

1. **Custom Plane Icon** – Your starter map shows an example of the default marker icon for Leaflet. Update your map to use one of the provided plane png icons as markers or choose one of your own.
2. **Rotate Plane Icon** – Your API data will include data relevant to the current direction the aircraft is flying relative to True North (0 degrees). Using the provided Leaflet Plugin (leaflet-rotatedmarker.js) Resource: <https://github.com/bbecquet/Leaflet.RotatedMarker> rotate each aircraft marker to indicate the direction it is travelling.
3. **Marker popups** – Leaflet provides the ability to load in data about each marker by leveraging a click event. You could fill this popup with some of the additional data provided by the API and stored as a Property in your geoJSON feature objects.

# Code Requirements

The following requirements are required for this assignment to meet the learning outcomes:

* **Your code must not contain loop structures of any kind.** Select from the available array functions that we’ve been exploring in order to accomplish the goals of the assignment.

# Code Submission

**You must commit and push all of your code to your provided GitHub repository.** It is also required that you commit and push often so that you build a history of commits and pushes that you can show to the instructor.

# Requirements (50 points)

1. **Demonstrate That you Have Retrieved the Required Raw Flight Data (20%)**

* Data must be filtered as specified earlier in the document
* You can demonstrate this by console.logging the raw data.

1. **Convert Raw Data into GeoJSON (20%)**

* Demonstrate transformation of filtered data into GeoJSON format.
* You can demonstrate this by console.logging the GeoJSON data.

1. **Plot Markers on Map to Show Each Aircraft (20%)**

* You can use the default marker for this requirement.

1. **Add Auto Refresh Functionality to the Page (20%)**

* API refreshes pretty reliably every seven (7) seconds.

1. **Additional Functionality (20%)**

* Plane Icon
* Icon Rotation
* Marker Popup with information about aircraft.

# Instructions

1. **Don’t forget that a live in-person demonstration of your code is part of this assignment. You will need to show your code to the instructor in class on the due date while going through an evaluation of your code’s functionality. Part of the assessment will include your ability to speak about the code you wrote, even if it doesn’t completely work or do what you expect.**
2. **Late submissions will be subject to the late penalties laid out in the course outline.**

# Academic Integrity and Plagiarism

**Code sharing by any means is considered plagiarism and is strictly forbidden under the NSCC Academic Integrity policy.**

[NSCC ACADEMIC INTEGRITY GUIDELINES](https://www.nscc.ca/docs/about-nscc/policies-procedures/policy-academicintegrity.pdf)

[NSCC ACADEMIC INTEGRITY REPORTING POLICY](https://www.nscc.ca/docs/about-nscc/policies-procedures/procedures-academicintegritystudent.pdf)