Module 14 Notes

**Looking Ahead**

In this module, you will use the Leaflet.js Application Programming Interface (API) to populate a geographical map with GeoJSON earthquake data from a URL. Each earthquake will be visually represented by a circle and color, where a higher magnitude will have a larger diameter and will be darker in color. In addition, each earthquake will have a popup marker that, when clicked, will show the magnitude of the earthquake and the location of the earthquake.

**What You Will Learn**

By the end of this module, you will be able to:

* Create a branch from the main branch on GitHub.
* Add, commit, and push data to a GitHub branch.
* Merge a branch with the main branch on GitHub.
* Retrieve data from a GeoJSON file.
* Make API requests to a server to host geographical maps.
* Populate geographical maps with GeoJSON data using JavaScript and the Data-Driven Documents (D3) library.
* Add multiple map layers to geographical maps using Leaflet control plugins to add user interface controls.
* Use JavaScript ES6 functions to add GeoJSON data, features, and interactivity to maps.
* Render maps on a local server.

Basil head of the reporting team; first team meeting, Basil assigns Sadhana as your mentor on this project, since she has created a similar map for severe weather. When you get back to your desk, you and Sadhana will go over the basic project plan and what you'll need to do.

Times projects for the Center for strategic International Studies

Decisions that are working in this module you'll build

creating interactive maps using GeoJSON data

## Basic Project Plan

#### Purpose

The purpose of this project is to visually show the differences between the magnitudes of earthquakes all over the world for the last seven days.

#### Tasks

To complete this project, use a URL for GeoJSON earthquake data from the USGS website and retrieve geographical coordinates and the magnitudes of earthquakes for the last seven days. Then add the data to a map.

#### Approach

Your approach will be to use the JavaScript and the D3.js library to retrieve the coordinates and magnitudes of the earthquakes from the GeoJSON data. You'll use the Leaflet library to plot the data on a Mapbox map through an API request and create interactivity for the earthquake data.

Now that you have an overview of the project plan, let's set up a Mapbox account and get the API token you'll need to create geographical maps.

pk.eyJ1IjoidGlnZ2VydGF6MzY5IiwiYSI6ImNsZzZ0eWtjdjBndnAzbW84N3piZWFxYmIifQ.LFavGnA8pQe1tqJNAcabBQ

Follow these steps to create a branch:

1. Navigate to the repository folder on your computer.
2. In Terminal on macOS or Git Bash on Windows, type git pull or git pull origin main and press Enter.
3. Type git checkout -b Simple\_Leaflet\_Map and press Enter.
   * git checkout lets us navigate between branches.
   * -b indicates we are creating a new branch.
   * The name of the new branch follows -b.

After pressing Enter, your terminal or Git Bash should return the following:

After pressing Enter, your terminal or Git Bash should return the following:

Switched to a new branch 'Simple\_Leaflet\_Map'

Now we are in the Simple\_Leaflet\_Map branch. Confirm this by typing git branch and pressing Enter. The output in Terminal or Git Bash should look as follows, with an asterisk next to the branch name:

\* Simple\_Leaflet\_Map

main

At this point, the folder structure on your computer hasn't changed. The files you had in your main branch are now in your Simple\_Leaflet\_Map branch. Visually, the repository should look like the following:

Now that we can create a simple Leaflet map, we can plot data on the map. First, let's create a new branch. Sadhana suggests that we name this branch "Mapping\_Single\_Points" since we'll map single points.

**REWIND**

Follow these steps to create a branch off of the main branch:

1. Navigate to your repository on your computer.
2. Make sure you're on the main branch by typing: git branch
3. If you're not on the main branch, type: git checkout main
4. Pull the changes from the main branch by typing: git pull
5. Create a new branch by typing: git checkout -b [name\_of\_your\_new\_branch]

##Overview

(As captured from Data Bootcamp Module 14)

In this module, as assigned Data Analyst, I will populate a geographical map with GeoJSON earthquake data from a URL. Each earthquake will be visually represented by a circle and color, whereas a higher magnitude will have a larger diameter and will be darker in color. In addition, each earthquake will have a popup marker that, when clicked, will show the magnitude of the earthquake and the location of the earthquake.

By the end of this module, as assigned Data Analyst, I will complete the following::

* Create a branch from the main branch on GitHub.
* Add, commit, and push data to a GitHub branch.
* Merge a branch with the main branch on GitHub.
* Retrieve data from a GeoJSON file.
* Make API requests to a server to host geographical maps.
* Populate geographical maps with GeoJSON data using JavaScript and the Data-Driven Documents (D3) library.
* Add multiple map layers to geographical maps using Leaflet control plugins to add user interface controls.
* Use JavaScript ES6 functions to add GeoJSON data, features, and interactivity to maps.
* Render maps on a local server.

##Purpose

(This module is built around a project that mirrors a real-world scenario that would require data analysis and visualization. In module 14, I will practice analysis, visualization, and statistical skills by retrieving and analyzing severe weather, in particular, Earthquake related data.)

## Working with the Client (Basil) we have created a Basic Project Plan, to ensure this our project is successfully implemented.

## Basic Project Plan

#### Purpose

The purpose of this project is to visually show the differences between the magnitudes of earthquakes all over the world for the last seven days.

#### Tasks

To complete this project, use a URL for GeoJSON earthquake data from the USGS website and retrieve geographical coordinates and the magnitudes of earthquakes for the last seven days. Then add the data to a map.

#### Approach

My approach will be to use the JavaScript and the D3.js library to retrieve the coordinates and magnitudes of the earthquakes from the GeoJSON data. You'll use the Leaflet library to plot the data on a Mapbox map through an API request and create interactivity for the earthquake data.

##Deliverable 1: Add Tectonic Plate Data

1. Create a new folder on your Mapping\_Earthquakes repository and name it "Earthquake\_Challenge."
2. Copy the folders and files from your Earthquakes\_past7days branch and add them to the Earthquake\_Challenge folder. The folder should have this structure:

A screenshot of a computer

Description automatically generated with medium confidence

1. Download the tectonic\_plate\_starter\_logic.js file, add it to your js folder, and rename it challenge\_logic.js.
2. In Step 1, add a second layer group variable for the tectonic plate data.
3. In Step 2, add a reference to the tectonic plate data to the overlay object.

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

1. In Step 3, using d3.json() callback method, make a call to the tectonic plate data using the GeoJSON/PB2002\_boundaries.json data from this [GitHub repositoryLinks to an external site.](https://github.com/fraxen/tectonicplates) for the tectonic plate data. You’ll need to log into GitHub to access the GeoJSON data.

Graphical user interface, text

Description automatically generated

1. Inside the d3.json() method do the following:
   * Pass the tectonic plate data to the geoJSON() layer.
   * Style the lines with a color and weight that will make it stand out on all maps.
   * Add the tectonic layer group variable you created in Step 1 to the map, i.e., .addTo(tectonicPlates) and close the geoJSON() layer.
   * Next, add the tectonic layer group variable to the map, i.e, tectonicPlates.addTo(map).
   * Finally, close the d3.json() callback.

Text

Description automatically generated

1. Start your Python server and launch the index.html file and confirm that your map with the earthquake and tectonic plate data is similar to the following image.

Text

Description automatically generated

Map

Description automatically generated

##Deliverable 2: Add Major Earthquake Data

1. In Step 1, add a third layer group variable for the major earthquake data.

Text

Description automatically generated

1. In Step 2, add a reference to the major earthquake data to the overlay object.

A picture containing text

Description automatically generated

1. In Step 3, use the d3.json() callback method to make a call to the major earthquake data from the GeoJSON Summary Feed for [M4.5+ EarthquakesLinks to an external site.](https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/4.5_week.geojson) for the Past 7 Days.

Text

Description automatically generated

1. In Step 4, use the same parameters in the styleInfo() function that will make a call to the getColor() and getRadius() functions.

Text

Description automatically generated

1. In Step 5, change the getColor() function to use only three colors for the following magnitudes; magnitude less than 5, a magnitude greater than 5, and a magnitude greater than 6.

Text

Description automatically generated

1. In Step 6, use the same parameters from the preceding step in the getRadius() function.

Text

Description automatically generated

1. In Step 7, pass the major earthquake data into the GeoJSON layer and do the following with the geoJSON() layer:
   * Turn each feature into a circleMarker on the map
   * Style each circle with styleInfo() function
   * Create a popup for the circle to display the magnitude and location of the earthquake
   * Add the major earthquake layer group variable you created in Step 1 to the map, i.e., .addTo(majorEQ) and then close the geoJSON() layer.

Text

Description automatically generated

1. Then, add the major earthquake layer group variable to the map, i.e, majorEQ.addTo(map), and then close the d3.json() callback.
2. Start your Python server and launch the index.html file and confirm that your map with the two earthquake data sets and tectonic plate data is similar to the following image.

Graphical user interface, text, website

Description automatically generated

Map

Description automatically generated

##Deliverable 3: Add an Additional Map

Using your knowledge of JavaScript and Leaflet.js add a third map style to your earthquake map.

1. Using the options from the [Mapbox stylesLinks to an external site.](https://docs.mapbox.com/api/maps/" \l "styles" \t "_blank), add a third map style as a tile layer object to the challenge\_logic.js file.

Text

Description automatically generated

1. Add the map variable to the base layer object.

Text

Description automatically generated

1. Start your Python server and launch the index.html file and confirm that your map is similar to the following image, where there are three map styles, and displays the two earthquake data sets and the tectonic plate data.

Map

Description automatically generated