## Introduction

The USGS is committed to providing scientific data about natural hazards, the health of our ecosystems and environment, and the impacts of climate and land-use change. With the aim to educate the public and government organizations about the critical issues facing our planet, this project is set to become a powerful tool by harnessing the power of real-time earthquake data to create an interactive map, depicting seismic activity.

## APIs Used

USGS Earthquake Data (API\_quakes):

* URL: [USGS GeoJSON Feed](https://chat.openai.com/c/API_quakes).
* Purpose: Provides real-time earthquake data.
* Content: Earthquake data updated every 5 minutes.
* Access: Publicly provided by the USGS.
* Use in Code: Fetching and visualizing earthquake data on a map.

## Features

* Displays earthquake events as circle markers, whose sizes and colors are determined by the earthquake magnitude.
* Option to switch between multiple base maps (like street, satellite, light, and timeline).
* Tooltip and popup features to display detailed information about each earthquake marker.
* Tectonic fault lines are displayed.
* Includes a layer control for easy toggle between different features.
* Supports a timeline feature to visualize earthquake events over time.
* Step-by-step Walkthrough
* Can be hosted using python -m http.server at localhost:8000 or VS Code Open with Live Server at localhost:5500 or the next available port.

## Code Summary

The given code is a JavaScript script that utilizes the Leaflet.js library to build an interactive map application for visualizing earthquake data and tectonic fault lines. It fetches data from USGS (United States Geological Survey) API and another dataset for tectonic plates.

Variable Declarations

* API\_quakes, API\_plates, fault\_line\_url: URLs to fetch earthquake and tectonic plate data

Function Definitions

* createMap(): Takes earthquake layers, timeline layer, faultline layer, and legend as arguments and sets up the Leaflet map.
* lightMap, tlMap, streetmap, satmap, googleTerrain: Different types of base maps.
* baseMaps, overlayMaps: Holds base and overlay maps respectively.
* layersControl: Adds layer controls to map.
* Event listener for base layer change to toggle between timeline and other views.
* addPopupInfo(): Adds tool-tip and pop-up information to each earthquake marker.

Setting Up the Map

* Using Leaflet, a map is created, centered, and zoomed to specific coordinates.
* A Mapbox access token must be available in the static/js/config.js file.

API Calls & Data Processing

* Fetch earthquake data from API\_quakes and process the GeoJSON data. The code retrieves real-time earthquake information from the USGS API, showing seismic activity approximately over 2.5 on the Richter scale.
  + Earthquakes are displayed with tooltips for additional information and a timeline function to see data populate in a time sequence.
* Create a color scale for earthquake markers. Data is transformed into markers or circles, sized and colored according to the magnitude.
* Define bins for earthquake magnitude and create corresponding layers using L.geoJSON().
* timelineLayer: Create a timeline layer that shows earthquakes over time.
* Call createMap() to render the map with the prepared layers and legend

Map Initialization

* myMap: Initializes the map with a default center and zoom level.
* Fetch default tile layer and add it to the map.