Earthquake Data USGS API Data

1. Refer to the below Site for the API data schema
   https://earthquake.usgs.gov/earthquakes/feed/v1.0/geojson.php

2. Refer below steps to get data from the source.
   a. First, you have to get historical data [Last month's data]
      https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_month.geojson
   b. After that, you have to get data for each day[Below URL will pull data for the past day]
      https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_day.geojson

3. Once you get this data from the source, please perform the below steps
   a. First, get data from the source
   b. Using Spark flatten all the columns from the source.
      i. Flatten column names, IF you are having nested columns make them unnest it.
         Example:
         “test”:”ha”,
         “Feature”:[
         {“Type”:”abc”
         “Name”:”abc”},
         {“Type”:”pqr”
         “Name”:”pqr”}
         ]

         After flattening the above JSON File, I should get the below columns in my target table.
         test, feature_type, feature_name

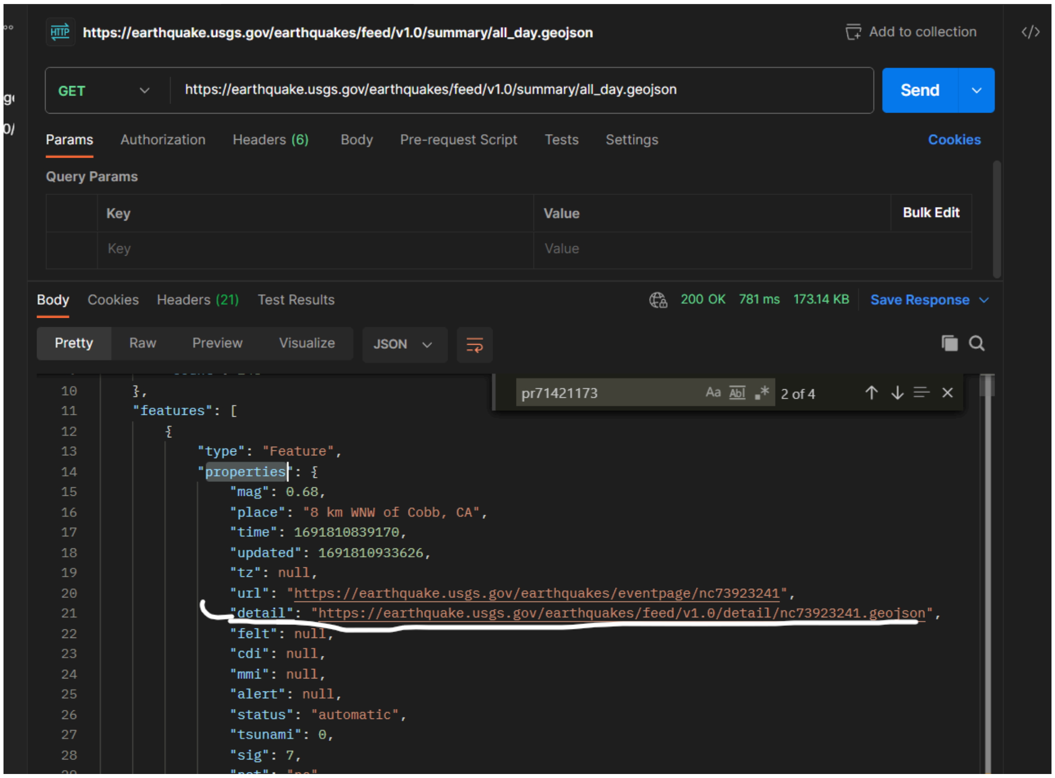
   c. and store them in the target location. Please refer target location
      earthquakeanalysis/raw/<date in YYYYMMDD>/<target file>.parquet

   d. In target data, you have the URL at the location,

      ```
      features.properties.detail
      ```

      Base url: https://earthquake.usgs.gov/
      End point url: earthquakes/feed/v1.0/detail/<id>.geojson

      Please refer below screenshot, for more details,



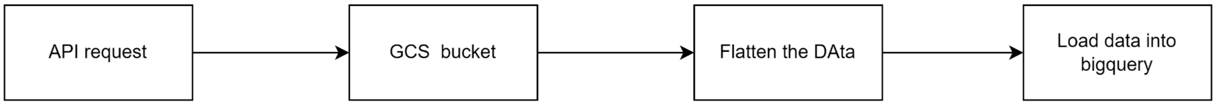
GET https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_day.geojson

```
    },
    "features": [
        {
            "type": "Feature",
            "properties": {
                "mag": 0.68,
                "place": "8 km WNW of Cobb, CA",
                "time": 1691810839170,
                "updated": 1691810933626,
                "tz": null,
                "url": "https://earthquake.usgs.gov/earthquakes/eventpage/nc73923241",
                "detail": "https://earthquake.usgs.gov/earthquakes/feed/v1.0/detail/nc73923241.geojson",
                "felt": null,
                "cdi": null,
                "mmi": null,
                "alert": null,
                "status": "automatic",
                "tsunami": 0,
                "sig": 7,
```

e. At this detail: location you have a URL , you have to pick this URL and pull data for this URL using rest API.
f. Using Pyspark you have to flatten all the columns in the data and store it in the below location earthquakeanalysis/raw/<date in YYYYMMDD>/<ids>_<target file>.parquet
g. Above highlighted yellow “ids” value you will get from the same URL or from the previously copied data.

4. Once this is done. To generate Analysis Layer Questions will be shared with you.

High level flow



Step 1: API Request

There are two scenarios

1. Using Pyspark - Dataproc or Databricks - python - request lib.
2. Using Cloud Dataflow - python request lib

Landing Location: gs://earthquake_analysis/pyspark/landing/20241019/*.json
gs://earthquake_analysis/dataflow/landing/20241019/*.json

Step 2. Flattening the data

1. Using Pyspark
2. Using Cloud dataflow

- While doing flattening also do below transformation
- Columns like “time”, “updated” - convert its value from epoch to timestamp
- Generate column “area” - based on existing “place” column

Silver Location: gs://earthquake_analysis/Silver/20241019/*.json

Flatten historical and daily data based on below example:

```
"mag": 0.89,
"place": "6 km NW of The Geysers, CA",
"time": 1729308248850,
"updated": 1729308343908,
"tz": null,
"url": "https://earthquake.usgs.gov/earthquakes/eventpage/nc75076006",
"detail": "https://earthquake.usgs.gov/earthquakes/feed/v1.0/detail/nc75076006.geojson",
"felt": null,
"cdi": null,
"mmi": null,
"alert": null,
"status": "automatic",
"tsunami": 0,
"sig": 12,
"net": "nc",
"code": "75076006",
"ids": ",nc75076006,",
"sources": ",nc,",
"types": ",nearby-cities,origin,phase-data,",
"nst": 9,
"dmin": 0.01303,
"rms": 0.02,
"gap": 77,
"magType": "md",
"type": "earthquake",
"title": "M 0.9 - 6 km NW of The Geysers, CA",
"geometry": {
```

```
"longtitude":-122.813163757324,
"latitude":38.8125,
"depth": 3.25999999046326
}
```

Step 3: Load data into Bigquery

- Add two extra columns
- 1. Insert data : insert_dt (Timestamp)

BQ Table: earthquake_db.earthquake_data

Do below Analysis using Pyspark and BigQuery

1. Count the number of earthquakes by region
2. Find the average magnitude by the region
3. Find how many earthquakes happen on the same day.
4. Find how many earthquakes happen on same day and in same region
5. Find average earthquakes happen on the same day.
6. Find average earthquakes happen on same day and in same region
7. Find the region name, which had the highest magnitude earthquake last week.
8. Find the region name, which is having magnitudes higher than 5.
9. Find out the regions which are having the highest frequency and intensity of earthquakes.

Cloud Composer

Historical load - Manual and its going to be one time activity
Daily Load -

- Ingestion - transformation - Bq load