```python
import pandas as pd
import json
import requests

GOOGLE_API_KEY = 'AIzaSyD3wR5ZmNZmv36w-LbmWixO5JKYmdWRtoI'
```

```python
data = pd.read_excel('data/company_data_full.xlsx')
```

```python
# Extract the headquarters locations for geocoding
headquarters = data[['headquarters_location']].drop_duplicates()
headquarters.value_counts()
data.columns
```

```
Index(['symbol', 'security', 'market_cap_weight(%)', 'market_cap_price',
       'sector', 'sub_industry', 'headquarters_location', 'headquarters_city',
       'headquarters_state', 'date_added', 'Des Moines'],
      dtype='object')
```

```python
api_list = []

def geocode_location(address_or_zipcode):
    lat, lng = None, None
    api_key = GOOGLE_API_KEY
    base_url = "https://maps.googleapis.com/maps/api/geocode/json"
    endpoint = f"{base_url}?address={address_or_zipcode}&key={api_key}"
    r = requests.get(endpoint)
    if r.status_code not in range(200, 299):
        return None, None
    try:
        results = r.json()['results'][0]
        api_list.append(results)
    except:
        print(f"Error: {r.json()}")
    return lat, lng
```

```python
# the list to append api
geometry = []
```

```
batch_size = 50
n_batches = len(headquarters) // batch_size + 1
# iterate over entire dataset
for i in range(n_batches):
    batch = headquarters.iloc[i*batch_size:(i+1)*batch_size]
    try:
        batch['latitude'], batch['longitude'] = zip(*batch['headquarters_location'].apply(ge
        geometry.append(batch)
    except:
        print(f"Error: {batch}")
    print(f"Batch {i+1}/{n_batches} completed")
```

```
Batch 1/6 completed
Batch 2/6 completed
Batch 3/6 completed
Batch 4/6 completed
Batch 5/6 completed
Batch 6/6 completed
```

```
def df_to_geojson(df, _properties, lat='latitude', lon='longitude'):
    geojson = {'type':'FeatureCollection', 'features':[]}
    for _, row in df.iterrows():
        feature = {'type':'Feature',
                   'properties':{},
                   'geometry':{'type':'Point',
                               'coordinates':[]}}
        feature['geometry']['coordinates'] = [row[lon],row[lat]]
        for prop in _properties:
            feature['properties'][prop] = row[prop]
        geojson['features'].append(feature)
    return geojson
```

```
display(geometry[0][:5])
```

	headquarters_location	latitude	longitude
0	Saint Paul, Minnesota	44.953703	-93.089958
1	Milwaukee, Wisconsin	43.038902	-87.906474
2	North Chicago, Illinois	42.325578	-87.841182
4	Dublin, Ireland	53.349805	-6.260310

	headquarters_location	latitude	longitude
5	San Jose, California	37.338740	-121.885253

```
# Define a function to extract city and state
def extract_city_state(location):
    parts = location.split(',')
    city = parts[0].strip()
    state = parts[1].strip() if len(parts) > 1 else None
    return city, state
```

```
geometry_df = pd.concat(geometry, ignore_index=True)

geometry_df[['city', 'state']] = geometry_df['headquarters_location'].apply(lambda x: pd.Seri

properties = ['city', 'state', 'latitude', 'longitude']

geojson_data = df_to_geojson(geometry_df, properties)
```

```
# Save the GeoJSON data to a file
with open('data/my_geometry.geojson', 'w') as f:
    json.dump(geojson_data, f)
```