In [1]:
```
from audioop import add
from xml.sax.handler import property_lexical_handler
from numpy import full
import requests
import pandas as pd
import re
import os.path
```

City and State inputs.

In [2]:
```
if 0:
    city = input("City?: ")
    if not re.match('^[A-Za-z]*$', city):
        print('Error! Only letters a-z allowed.')
    state = input(f'Two Character State?: ')
    if not re.match('^[A-Za-z]*$', state):
        print('Error! Only letters a-z allowed.')
    elif len(state) > 2:
        print('Error! Only 2 Characters allowed!')
    print(f'Looking for houses in {city}, {state}.')
else:
    city = 'Beaverton'
    state = 'OR'
```

In [3]:
```
agent = []
home_type = []
year_built = []
address = []
bedrooms = []
bathrooms = []
sq_foot = []
price = []
sold_date = []
sold_price = []
```

In [ ]:
```
offset = 0
page_len = 200
total = 1
count = 1
```

In [ ]:
```
while page_len > 0:
    page_increment = '/pg_1'
    pageindex = f'https://www.realtor.com/realestateandhomes-search/{city}_{state}'
    headers = {
        'accept': 'application/json',
    }
    params = {
        'client_id': 'rdc-x',
        'schema': 'vesta',
    }
    json_data = {
        # This is a graphql query, so you can change what data you get back
        'query': '''
    query ConsumerSearchMainQuery($query: HomeSearchCriteria!, $limit: Int, $offset: I
    {
```

```
    home_search: home_search(query: $query,
        sort: $sort,
        limit: $limit,
        offset: $offset,
        sort_type: $sort_type,
        client_data: $client_data,
        bucket: $bucket,
    ){
        count
        total
        results {
        property_id
        list_price
        listing_id
        matterport
        status
        permalink
        price_reduced_amount
        description{
        beds
        baths
        baths_full
        baths_half
        baths_1qtr
        baths_3qtr
        garage
        stories
        type
        sub_type
        lot_sqft
        sqft
        year_built
        sold_price
        sold_date
        name
        }
        location{
        street_view_url
        address{
            line
            postal_code
            state
            state_code
            city
            coordinate {
            lat
            lon
            }
        }
        county {
            name
            fips_code
        }
        }
    }
    }
    }''',
        'variables': {
            'query': {
                'status': [
                    'for_sale',
                    'ready_to_build',
                ],
                'primary': True,
                'search_location': {
                    'location': (f'{city}, {state}'),
                },
            },
            'client_data': {
                'device_data': {
                    'device_type': 'web',
                },
                'user_data': {
                    'last_view_timestamp': -1,
                },
            },
            'limit': page_len,
            'offset': offset,
            'zohoQuery': {
                'silo': 'search_result_page',
                'location': (f'{city}'),
                'property_status': 'for_sale',
                'filters': {},
                'page_index': (f'{page_increment}'),
            },
            'geoSupportedSlug': (f'{city}_{state}'),
            'sort': [
                {
                    'field': 'list_date',
                    'direction': 'desc',
                },
                {
                    'field': 'photo_count',
                    'direction': 'desc',
                },
            ],
            'by_prop_type': [
                'home',
            ],
        },
        'operationName': 'ConsumerSearchMainQuery',
        'callfrom': 'SRP',
        'nrQueryType': 'MAIN_SRP',
        'visitor_id': '1ae3a798-c7a2-4fd6-bc2a-b84aec36420f',
        'isClient': True,
        'seoPayload': {
            'asPath': (f'{pageindex}{page_increment}'),
            'pageType': {
                'silo': 'search_result_page',
                'status': 'for_sale',
            },
            'county_needed_for_uniq': False,
        },
    }

    response = requests.post('https://www.realtor.com/api/v1/hulk_main_srp', params=pa
```

json object

```
In [ ]:     result_items = response.json()['data']['home_search']
            page_len = result_items['count']
            offset += page_len
```

result items

```
In [ ]: result_items = result_items['results']
```

```
In [ ]: for result in result_items:
                try:
                    agent.append(result['Branding']['0']['name'])
                except:
                    agent.append('')
                try:
                    home_type.append(result['description']['type'])
                except:
                    home_type.append('')
                try:
                    year_built.append(result['description']['year_built'])
                except:
                    year_built.append('')
                try:
                    address.append(result['location']['address']['line'])
                except:
                    address.append('')
                try:
                    bedrooms.append(result['description']['beds'])
                except:
                    bedrooms.append('')
                try:
                    bathrooms.append(result['description']['baths'])
                except:
                    bathrooms.append('')
                try:
                    sq_foot.append(result['description']['lot_sqft'])
                except:
                    sq_foot.append('')
                try:
                    price.append(result['list_price'])
                except:
                    price.append('')
                try:
                    sold_date.append(result['description']['sold_date'])
                except:
                    sold_date.append('')
                try:
                    sold_price.append(result['description']['sold_price'])
                except:
                    sold_price.append('')
```

```
In [ ]: df_realtor = pd.DataFrame({'Agent': agent, 'Home Type': home_type, 'Year Built': year_
```

```
In [ ]: df_realtor
```

Cross-platform filepath

```
In [ ]: fname = os.path.join('csv', (f'realtor_data_{city}_{state}.csv'))
        df_realtor.to_csv(fname, header=True)
```