

```

In [ ]: import requests
import time
import pandas as pd
url = 'https://api.lens.org/scholarly/search'

# include fields
include = '''["patent_citation_count", "publication_type",
"publication_supplementary_type", "author.collective_name",
"author.first_name", "author.last_name", "author.display_name",
"author.affiliation.name", "author.affiliation.name.exact",
"author.affiliation.grid_id", "author.affiliation.country_code",
"title", "language", "field_of_study", "abstract",
"date_published", "year_published", "conference.name",
"author_count", "reference_count", "scholarly_citation_count",
"source.title", "source.title.exact", "source_type",
"source.publisher", "source.country", "keyword",
"funding.organisation", "funding.organisation.exact", "funding.country"]'''
# request body with scroll time of 1 minute
request_body = '''{
    "query": "(author.affiliation.country_code:GH AND year_published:[2000 T
    "size": 100,
    "scroll": "1m",
    "include": %s
}''' % include
headers = {'Authorization': 'AodFdjadfkZP1nzTs3Tf7bJdNnFthJ9ZTLeeAIZKLFM4MCo7

# Recursive function to scroll through paginated results
def scroll(scroll_id):
    # Change the request_body to prepare for next scroll api call
    # Make sure to append the include fields to make faster response
    if scroll_id is not None:
        global request_body
        request_body = '''{"scroll_id": "%s", "include": %s}''' % (scroll_id,

# make api request
response = requests.post(url, data=request_body, headers=headers)

# If rate-limited, wait for n seconds and proceed the same scroll id
# Since scroll time is 1 minutes, it will give sufficient time to wait and
if response.status_code == requests.codes.too_many_requests:
    time.sleep(4)
    scroll(scroll_id)
# If the response is not ok here, better to stop here and debug it
else:
    json = response.json()
    scroll_id = json['scroll_id'] # Extract the new scroll id from respon
    if len(json['data']) == 0:
        return [json]
    return [json] + scroll(scroll_id)

# start recursive scrolling
json = scroll(scroll_id=None)
json = json[:-1]

```

```
In [ ]: def convert_to_df(json):
        scroll_id = []
        total = []
        data = []
        results = []
        for j in json:
            scroll_id.append(j["scroll_id"])
            total.append(j["total"])
            data.append(j["data"])
            results.append(j["results"])
        df = pd.DataFrame(
            {'scroll_id': scroll_id,
            'total': total,
            'data': data,
            'results': results
            }
        )
        return df
df = convert_to_df(json)
```

```
In [ ]: df['data'][0]
```

```
In [ ]: df.head()
```

```
In [ ]: new_df = pd.DataFrame(df['data'][0])
        for i in range(1, len(df)):
            new_df = new_df.append(pd.DataFrame(df['data'][i]), ignore_index = True)
```

```
In [ ]: new_df.head()
```

```
In [9]: new_df.to_csv('GH2000.csv')
```

```
In [ ]:
```