

# Final Project - Data Collection Step -

## Information Retrieval Using 'Event Registry' Searching Platform

Freda Xiaoyun Yu

UOL number: 190178194

Date: 22 Oct 2022

This file contains the records I have operated when I retrieve the articles data from the 'Event Registry' website. The API they provide is called 'NewsAPI', which can be found at <https://www.newsapi.ai/documentation/sandbox?tab=searchArticles> (<https://www.newsapi.ai/documentation/sandbox?tab=searchArticles>)

First, I need an account. Then, from my account I find out my API key. This should be filled in the codes below.

To install the Event Registry API,

- I am following the guidelines here <https://github.com/EventRegistry/event-registry-python/wiki> (<https://github.com/EventRegistry/event-registry-python/wiki>)
- Download the eventregistry-8.12.tar.gz from <https://pypi.org/simple/eventregistry/> (<https://pypi.org/simple/eventregistry/>) to local folder C:\Users\Administrator\AppData\Roaming\Python
- Close the Astrill VPN, because with this VPN, system will not find the correct path.
- In the local folder, press Shift + RightClick, to open PowerShell at this location
- Based on the notification, python.exe -m pip install --upgrade pip
- PS C:\Users\Administrator\AppData\Roaming\Python> pip install eventregistry-8.12.tar.gz
- Run Python3.8, import eventregistry
- In the PowerShell, we may pip install eventregistry --upgrade
- C:\Python\Python38\Lib\site-packages\eventregistry In this directory, create a new JSON file called 'settings.json' and add:

```
{
  "apiKey": "5d9ca3e2-04a6-4cf5-b47c-4346d55c385a"
}
```

- Open Jupyter Notebook. Create a new Python file. Import the module:

```
import eventregistry
```

```
from eventregistry import *
```

- Open the website <https://www.newsapi.ai/documentation/sandbox?tab=searchArticles> (<https://www.newsapi.ai/documentation/sandbox?tab=searchArticles>) to try to use the sandbox to retrieve results. Copy the Python codes from the code block, to my Jupyter Notebook file.

In [ ]:

In [ ]:

```
# https://github.com/EventRegistry/event-registry-python/wiki
```

In [1]:

```
import eventregistry
from eventregistry import *
```

In [14]:

```

from eventregistry import *
# since we want recent results, just prevent use of archive - in this way we don't need to set any date constraints
er = EventRegistry(apiKey = '5d9ca3e2-04a6-4cf5-b47c-4346d55c385a', allowUseOfArchive = False)

q = QueryArticlesIter(keywords = ["Barack Obama"], dataType = ["news", "pr"])
# we limit here the results to 100. If you want more, remove or increase maxItems
for article in q.execQuery(er, sortBy="date", sortByAsc=False, maxItems=100):
    print(article)

# alternatively, the same query using the query language could look something like this:
qStr = """
{
  "$query": {
    "keyword": "Barack Obama"
  },
  "$filter": {
    "endSourceRankPercentile": 30,
    "dataType": ["news", "pr"]
  }
}
"""
q = QueryArticlesIter.initWithComplexQuery(qStr)

```

```

{'uri': '7234568518', 'lang': 'eng', 'isDuplicate': False, 'date': '2022-10-22', 'time': '06:23:00', 'dateTime': '2022-10-22T06:23:00Z', 'dateTimePub': '2022-10-22T05:37:00Z', 'dataType': 'news', 'sim': 0.7450980544090271, 'url': 'https://dcweekly.org/2022/10/22/republican-senate-candidate-adam-laxalt-surging-in-nevada-now-leading-in-rasmussen-poll/', 'title': 'Republican Senate Candidate Adam Laxalt Surging In Nevada, Now Leading In Rasmussen Poll', 'body': 'The Nevada senate race is looking very good for Republicans as Adam Laxalt is now polling better than Democrat incumbent Catherine Cortez Masto.\\n\\nThis is one of the races that Democrats have been worried about for a few weeks now and with good reason.\\n\\nNevada would be an important pickup for Republicans in the battle for the Senate.\\n\\nPolitico reports:\\n\\nA bad sign for Democrats in critical Nevada Senate race\\n\\nOne of the tightest Senate races in the country has drawn even tighter in the final weeks of the midterm election season, with Republican Adam Laxalt now polling even with Democratic Catherine Cortez Masto in Nevada -- a race that is one of the GOP's best shots at flipping a Democratic-held seat.\\n\\nLaxalt has inched ahead of Cortez Masto by 2 percentage points, within the poll's margin of error, a gain from a month ago when he was down 3 percentage points, according to a poll conducted this week by the conservative Club for Growth and shared exclusively with POLITICO.\\n\\nThe bump for Laxalt represents a swing toward Republicans as concerns about the economy loom large in contrast with a Democratic summer boost in momentum over abortion rights. Independent voters appear to be breaking with the GOP as the Nov. 8 election nears.\\n\\nLaxalt's slight lead in public polls comes as Democrats have called in top surrogates to appear in Nevada in the next two weeks, including former President Barack Obama and Sen. Bernie Sanders. Earlier this month, former President Donald Trump held a rally to support Laxalt.', 'source': {'uri': 'dcweekly.org', 'dataType': 'news', 'title': 'DC Weekly'}, 'authors': [], 'image': 'https://dcweekly.org/wp-content/uploads/2022/10/Adam-Laxalt-NV-dtW1X3.jpeg', 'eventUri': 'eng-8113941', 'sentiment': 0.1843137254901961, 'wgt': 404115780, 'relevance': 1}

```

In [ ]:

```

# Above is just a trial, with codes shown in their examples.
# Below, I will write my own retrieval codes.

```

In [ ]:

In the sandbox website <https://www.newsapi.ai/documentation/sandbox?tab=searchArticles> (<https://www.newsapi.ai/documentation/sandbox?tab=searchArticles>), settle the below choices:

## Get articles

Use this API endpoint to find articles that match one or more search conditions.

## Input parameters

What topics are you interested in?

Filters: 📍 Locations 📊 Science/Top 15 📁 dmoz:Science→Physics 📅 From 2021-01-01 🌐 English ⚙ Misc

Sort results by: Date

Return type: List of articles

Nr. of returned articles: 100

Returned info for: Article Concept Source

5 selected Label Title

- ✓ Basic information
- ✓ Title
- ✓ Body
- ✓ Event URI
- Social score
- Concepts
- Categories
- Location
- Image
- Videos
- Extracted links
- Extracted dates
- Duplicate list
- ✓ Original article

EXECUTE SEARCH

- Change the final printing sentence to a 'write-to-file' sentence. We need to create a new JSON file in the directory we want to put it in, and set the mode as 'appending', meaning we do not delete the original contents:

In [4]:

```
import json

er = EventRegistry(apiKey = '5d9ca3e2-04a6-4cf5-b47c-4346d55c385a')
qStr = """
{
  "$query": {
    "$and": [
      {
        "categoryUri": "dmoz/Science/Earth_Sciences"
      },
      {
        "sourceGroupUri": "science/top15"
      },
      {
        "lang": "eng"
      }
    ]
  },
  "$filter": {
    "forceMaxDateTimeWindow": "31",
    "dataType": [
      "news"
    ]
  }
}
"""

q = QueryArticlesIter.initWithComplexQuery(qStr)
# change maxItems to get the number of results that you want
for article in q.execQuery(er, maxItems=100):
    with open('D:/University-of-London-2020/CM3070-Computer-Science-Final-Project/datasets/Event_registry_search_results/API/output.json', 'a') as f:
        f.write(json.dumps(article, ensure_ascii=True))
```

In [5]:

```
import json

er = EventRegistry(apiKey = '5d9ca3e2-04a6-4cf5-b47c-4346d55c385a')
qStr = """
{
    "$query": {
        "$and": [
            {
                "categoryUri": "dmoz/Science/Physics"
            },
            {
                "sourceGroupUri": "science/top15"
            },
            {
                "lang": "eng"
            }
        ]
    },
    "$filter": {
        "forceMaxDataTimeWindow": "31",
        "dataType": [
            "news"
        ]
    }
}
"""

q = QueryArticlesIter.initWithComplexQuery(qStr)
# change maxItems to get the number of results that you want
for article in q.execQuery(er, maxItems=100):
    with open('D:/University-of-London-2020/CM3070-Computer-Science-Final-Project/datasets/Event_registry_search_results/API/physics.json', 'a') as f:
        f.write(json.dumps(article, ensure_ascii=True))
```

In [6]:

```
import json

er = EventRegistry(apiKey = '5d9ca3e2-04a6-4cf5-b47c-4346d55c385a')
qStr = """
{
    "$query": {
        "$and": [
            {
                "categoryUri": "dmoz/Science/Biology"
            },
            {
                "sourceGroupUri": "science/top15"
            },
            {
                "lang": "eng"
            }
        ]
    },
    "$filter": {
        "forceMaxDataTimeWindow": "31",
        "dataType": [
            "news"
        ]
    }
}
"""

q = QueryArticlesIter.initWithComplexQuery(qStr)
# change maxItems to get the number of results that you want
for article in q.execQuery(er, maxItems=100):
    with open('D:/University-of-London-2020/CM3070-Computer-Science-Final-Project/datasets/Event_registry_search_results/API/biology.json', 'a') as f:
        f.write(json.dumps(article, ensure_ascii=True))
```

In [7]:

```
import json

er = EventRegistry(apiKey = '5d9ca3e2-04a6-4cf5-b47c-4346d55c385a')
qStr = """
{
    "$query": {
        "$and": [
            {
                "categoryUri": "dmoz/Science/Math"
            },
            {
                "sourceGroupUri": "science/top15"
            },
            {
                "lang": "eng"
            }
        ]
    },
    "$filter": {
        "forceMaxDateTimeWindow": "31",
        "dataType": [
            "news"
        ]
    }
}
"""

q = QueryArticlesIter.initWithComplexQuery(qStr)
# change maxItems to get the number of results that you want
for article in q.execQuery(er, maxItems=100):
    with open('D:/University-of-London-2020/CM3070-Computer-Science-Final-Project/datasets/Event_registry_search_results/API/math.json', 'a') as f:
        f.write(json.dumps(article, ensure_ascii=True))
```

In [8]:

```
import json

er = EventRegistry(apiKey = '5d9ca3e2-04a6-4cf5-b47c-4346d55c385a')
qStr = """
{
    "$query": {
        "$and": [
            {
                "categoryUri": "dmoz/Science/Social_Sciences"
            },
            {
                "sourceGroupUri": "science/top15"
            },
            {
                "lang": "eng"
            }
        ]
    },
    "$filter": {
        "forceMaxDateTimeWindow": "31",
        "dataType": [
            "news"
        ]
    }
}
"""

q = QueryArticlesIter.initWithComplexQuery(qStr)
# change maxItems to get the number of results that you want
for article in q.execQuery(er, maxItems=100):
    with open('D:/University-of-London-2020/CM3070-Computer-Science-Final-Project/datasets/Event_registry_search_results/API/social_sciences.json', 'a') as f:
        f.write(json.dumps(article, ensure_ascii=True))
```

```
In [9]:

import json

er = EventRegistry(apiKey = '5d9ca3e2-04a6-4cf5-b47c-4346d55c385a')
qStr = """
{
    "$query": {
        "$and": [
            {
                "categoryUri": "dmoz/Science/Chemistry"
            },
            {
                "sourceGroupUri": "science/top15"
            },
            {
                "lang": "eng"
            }
        ]
    },
    "$filter": {
        "forceMaxDataTimeWindow": "31",
        "dataType": [
            "news"
        ]
    }
}
"""

q = QueryArticlesIter.initWithComplexQuery(qStr)
# change maxItems to get the number of results that you want
for article in q.execQuery(er, maxItems=100):
    with open('D:/University-of-London-2020/CM3070-Computer-Science-Final-Project/datasets/Event_registry_search_results/API/chemistry.json',
              'a') as f:
        f.write(json.dumps(article, ensure_ascii=True))
```

Till now, we should already have had JSON files newly created in the director we have specified, within each there are 100 articles with detailed information.

My directory now look like this (I have renamed the file 'output.json' into 'earth-sciences.json'):

datasets > Event_registry_search_results > API					在 API 中搜索	
	名称	修改日期	类型	大小		
同步	my_trial	2022/10/22 星期...	文件夹			
	biology.json	2022/10/22 星期...	JSON 文件	1,525 KB		
	chemistry.json	2022/10/22 星期...	JSON 文件	1,001 KB		
	math.json	2022/10/22 星期...	JSON 文件	1,139 KB		
	output.json	2022/10/22 星期...	JSON 文件	647 KB		
	physics.json	2022/10/22 星期...	JSON 文件	757 KB		
	social_sciences.json	2022/10/22 星期...	JSON 文件	817 KB		

the JSON file should look like this:

```
1 {"uri": "7213732540", "lang": "eng", "isDuplicate": false, "date": "2022-10-05",
  "time": "17:00:00", "dateTime": "2022-10-05T17:00:00Z", "dateTimePub": "2022-10-05T17:00:00Z", "dataType": "news", "sim": 0.6705882549285889, "url":
  "https://www.nature.com/collections/ddgjcjgddi", "title": "Nobel Prize in Chemistry
  2022", "body": "The 2022 Nobel Prize in Chemistry has been awarded to Carolyn R.
  Bertozzi, Morten Meldal and K. Barry Sharpless for the development of click chemistry
  and bioorthogonal chemistry. Click chemistry - independently reported in 2002 by
  Meldal and Sharpless - reacts two molecules together in a simple and efficient way.
  The practical and reliable nature of click reactions swiftly made them popular for
  the synthesis of small well-defined molecules as well as extended materials such as
  polymers. The utility of the click chemistry concept was successfully demonstrated
  beyond the reaction flask by Bertozzi, who showed that it could be applied in living
  systems to, for example, probe glycans on cell surfaces. By designing these reactions
  to be biocompatible they do not interfere with the chemistry of natural systems and
  in this way has underpinned the burgeoning field of bioorthogonal chemistry.\n\nIn
  this Collection, Nature Portfolio recognizes the achievements of the Laureates in a
  selection of research, review, news and opinion articles that highlight the
  development of click chemistry and bioorthogonal chemistry over the past two
  decades.", "source": {"uri": "nature.com", "dataType": "news", "title": "Nature"},
  "authors": [], "image": "https://media.springernature.com/w110/nature-
  cms/uploads/collections/Nobel_Chemistry_Hero-
  454761d265ae552b6c1ccae4fdde75d5_92_-454761d265ae552b6c1ccae4fdde75d5.jpg",
  "eventUri": "eng-8076924", "sentiment": 0.5137254901960784, "wgt": 21, "relevance":
  21}{ "uri": "7231239765", "lang": "eng", "isDuplicate": false, "date": "2022-10-19",
  "time": "15:56:00", "dateTime": "2022-10-19T15:56:00Z", "dateTimePub": "2022-10-
  19T15:56:00Z", "dataType": "news", "sim": 0, "url":
```

Now it's my task to transform the JSON file into CSV or EXCEL format.

In [ ]:

Now since 100 articles for each category may not be sufficient, I will retrieve more data, ideally another 100 for each previous category. I will set the time period as 2018-01-01 to 2020-12-31, since previously I have settled the time period as 2021-01-01 to present.

In [10]:

```
import json

er = EventRegistry(apiKey = '5d9ca3e2-04a6-4cf5-b47c-4346d55c385a')
qStr = """
{
    "$query": {
        "$and": [
            {
                "categoryUri": "dmoz/Science/Chemistry"
            },
            {
                "sourceGroupUri": "science/top15"
            },
            {
                "lang": "eng"
            }
        ]
    },
    "$filter": {
        "forceMaxDataTimeWindow": "31",
        "dataType": [
            "news"
        ]
    }
}
"""

q = QueryArticlesIter.initWithComplexQuery(qStr)
# change maxItems to get the number of results that you want
for article in q.execQuery(er, maxItems=100):
    with open('D:/University-of-London-2020/CM3070-Computer-Science-Final-Project/datasets/Event_registry_search_results/API/chemistry2.json',
              'a') as f:
        f.write(json.dumps(article, ensure_ascii=True))
```

However, after downloading this second file, I have found from the text-comparison website

<https://app.copyleaks.com/dashboard/v1/report/bq5pe67t64721gjl/preview?key=za30mksexzmbuukl&suspectId=ad89af9002&viewMode=one-to-one&contentMode=html&sourcePage=1&suspectPage=1> (https://app.copyleaks.com/dashboard/v1/report/bq5pe67t64721gjl/preview?key=za30mksexzmbuukl&suspectId=ad89af9002&viewMode=one-to-one&contentMode=html&sourcePage=1&suspectPage=1) that 96% of the contents are the same, which means we cannot use this new file as new data.

In [ ]:

Till now, I have utilised the EventRegistry API to successfully retrieve the datasets that I want.

In [ ]:

In [ ]: