**PUBMED QUERY API**

**Python Libraries Required:**

**Python version: 3.8>**

* Pandas: Data Manipulation
* Poetry: Dependency management
* Requests: Hnadles HTTP REQUEST
* XML: Handles Xml Data
* CSV

**PUBMED REST APIS :**

1.[**https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&term=science[journal]+AND+breast+cancer+AND+2008[pdat]&usehistory=y**](https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&term=science%5bjournal%5d+AND+breast+cancer+AND+2008%5bpdat%5d&usehistory=y)

**2.**[**https://esummary.fcgi?db=pubmed&id=<uidt**](https://esummary.fcgi?db=pubmed&id=%3cuidt)**>**

**SCRIPT FLOW**

* **Pubmed python script SENDS GET request to the 1st endpoint with UserQuery**
* **Receives response in xml format**
* **Scrapes the UID of Top document**
* **Sends GET request to 2nd Endpoint with UID**
* **Receives response in xml format**
* **Scrapes Required Metadata from XML RESPONSE and generates CSV File**

**SOURCECODE:**

import requests

import xml.etree.ElementTree as ET

import csv

from datetime import datetime

# Step 1: Get the search query from the user

query = str(input("Enter your search query: "))

ufilename = ""

# Command options

opt = str(input("Enter option (-h for help, -f for filename input): "))

if opt == "-h":

    print("1. Enter the Book name for book metadata.")

    print("2. Default metadata is generated with a timestamped CSV file or provide a file name using -f.")

elif opt == "-f":

    ufilename = str(input("Enter your custom file name: "))

# Construct the URL for the esearch endpoint

search\_url = f'https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed&term={query}'

# Step 2: Make the API request to esearch

response = requests.get(search\_url)

# Check if the request was successful

if response.status\_code == 200:

    # Parse the XML response from the esearch endpoint

    data = ET.fromstring(response.text)

    # Extract total count of results and list of PubMed IDs

    count = data.find("Count").text if data.find("Count") is not None else "0"

    id\_list = [id\_elem.text for id\_elem in data.findall("IdList/Id")]

    # Get the top PubMed ID (if available)

    if id\_list:

        top\_pmid = id\_list[0]

        print(f"Top PubMed ID: {top\_pmid}")

        # Step 3: Fetch detailed document info using esummary with the top PubMed ID

        esummary\_url = f'https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esummary.fcgi?db=pubmed&id={top\_pmid}&retmode=xml'

        esummary\_response = requests.get(esummary\_url)

        if esummary\_response.status\_code == 200:

            # Parse the XML response from the esummary endpoint

            esummary\_data = ET.fromstring(esummary\_response.text)

            # Extract relevant data from the XML response

            doc = esummary\_data.find("DocSum")

            title = doc.find("Item[@Name='Title']").text if doc.find("Item[@Name='Title']") is not None else "N/A"

            pub\_date = doc.find("Item[@Name='PubDate']").text if doc.find("Item[@Name='PubDate']") is not None else "N/A"

            authors = [author.text for author in doc.findall("Item[@Name='Author']")]

            affiliations = [affil.text for affil in doc.findall("Item[@Name='Affil']", namespaces={'': ''})]  # Namespace handling

            email = doc.find("Item[@Name='Email']").text if doc.find("Item[@Name='Email']") is not None else "N/A"

            # Use the custom filename if provided, otherwise generate a timestamped filename

            if ufilename != '':

                filename = ufilename

            else:

                filename = f"fetched{datetime.now().strftime('%Y%m%d%H%M%S')}.csv"

            # Step 4: Write the extracted data into a CSV file

            with open(filename, mode='w', newline='') as file:

                writer = csv.writer(file)

                # Write the header

                writer.writerow(['Title', 'Publication Date', 'Authors', 'Affiliations', 'Author Email'])

                # Write the article details

                writer.writerow([title, pub\_date, ', '.join(authors), ', '.join(affiliations), email])

            print(f"Data saved to {filename}")

        else:

            print(f"Error fetching document details: HTTP {esummary\_response.status\_code}")

    else:

        print("No PubMed IDs found.")

else:

    print(f"Error: HTTP {response.status\_code}")