**Documentation II**

This second documentation specifically talks about our LinkedIn connection and how we implement simple scraping of Google Scholar. All codes are included in the ZIP file this documentation came in with. By the time this project was created, the LinkedIn code is hosted on <https://amazingteam.site/access.php>

* **LinkedIn**

**Installation**

Basic need:

* A company verified account access of LinkedIn. Simply put, you can not use the API if you are not a legitimate employee of a said app. We worked this around by registering our own company in LinkedIn, a simple one-man company would do. Just make sure that your created company is listed on LinkedIn searches. Or, simply use our client key and client id, that way you don’t need a LinkedIn developer account.
* A developer account, you can do this by going to <https://www.linkedin.com/developers/> . That said, you can follow the official guide from there. Then create the app of your company from earlier.
* A basic php server online, with a legitimate domain. You can either set it up or buy a domain and host service. LinkedIn Requires the API user to have these to use as a call-back URL, more on that later.

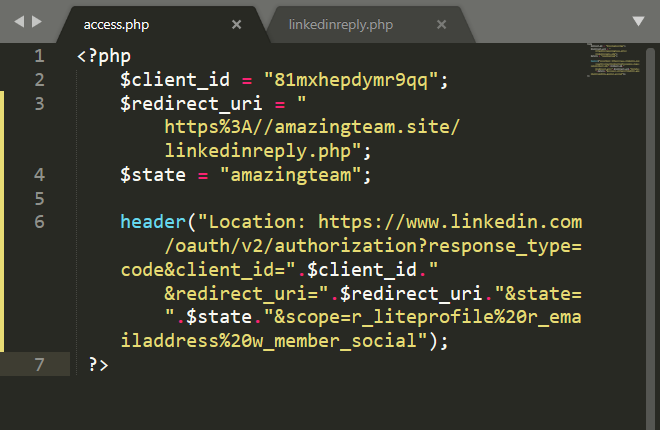
A quick look at the code:

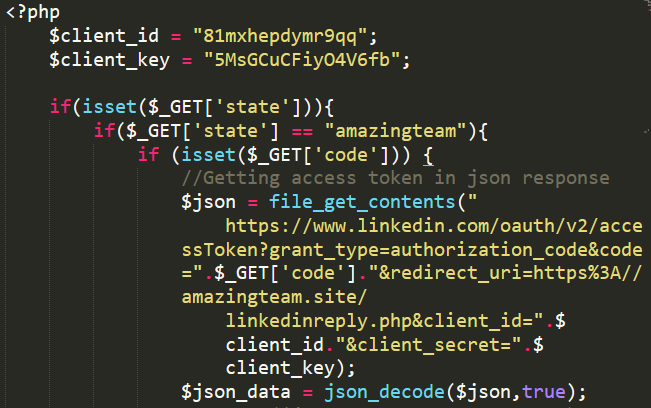
To put it simply, there are three authentication steps to get the user data of LinkedIn:

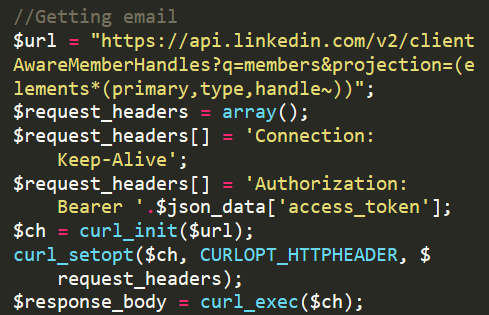
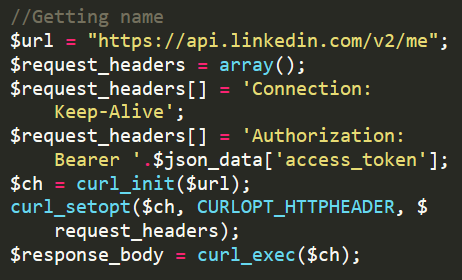
* + Verify yourself as a legitimate app
  + Asked the user to agree
  + Exchange the user agree consent with access token

There will be 2 files of php with this project:

* + access.php



* + - Send your app key (app dev page in basic need number 2)
    - Redirect the user to a LinkedIn authorize page
  + linkedinreply.php
    - Get the authorization code from the page authorize page
    - Exchange the auth code with actual access token
    - Use this access token to get user data



Keep in mind:

* The access token has an expiration date, usually it is 90 days.
* Your client id and client key is unique to your app. You can however use our client key and id for debugging purposes.
* To get more data aside from basic profile and email, your company whose app you use has to have an acknowledgement from LinkedIn.
* The end reply from LinkedIn is a JSON file with user data inside.

**Google Scholar**

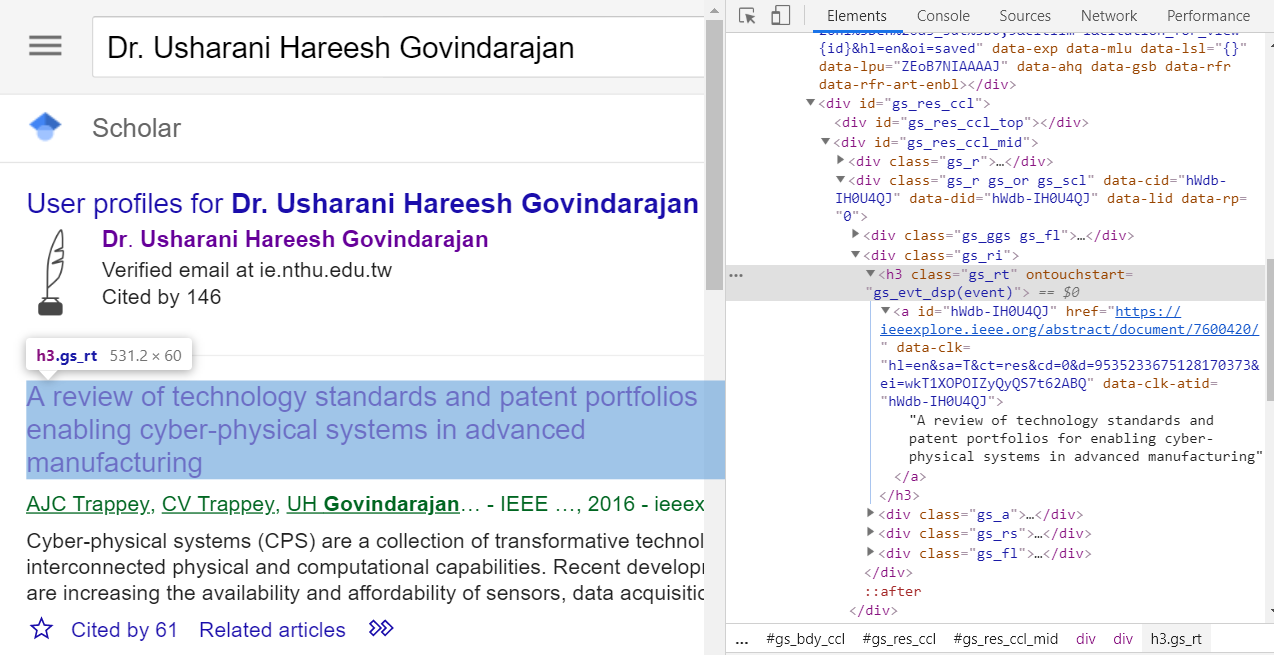
Environment used:

* Anaconda with Anaconda Navigator with Python 3.7 installed,
* Edited with Jupyter Lab
* Python library BeautifulSoup

Steps in code:

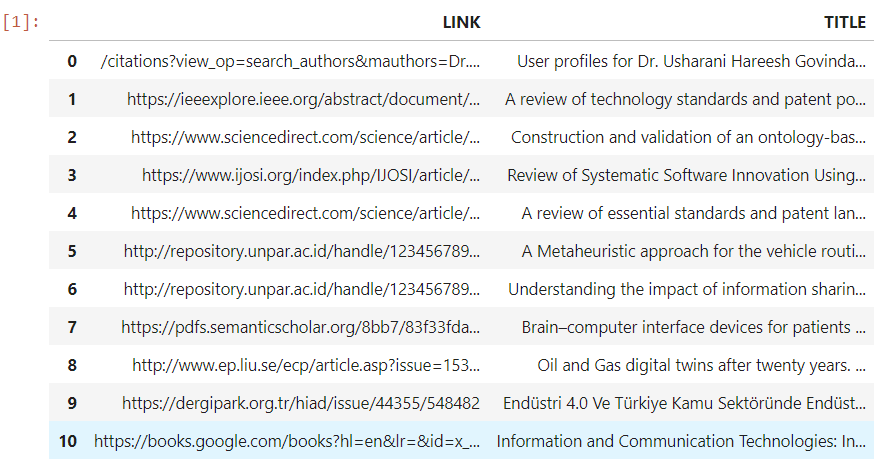
* Determine search query
* Prepare http URL
* Determine which element you want to take
* Parse the webpage to BeautifulSoup

A quick look at what attribute you want to get, in this case it is ‘h3.gs\_rt’ since it contains the title of the research





The result may look something like this



Keep in mind:

* The query is a crude one, by following a simple search term, so you must use http string request pattern like the example above

Trivia:

* By the time the project was created, there were already a library called ‘scholarly’ for Python. It provides a much easier way to scrape the data. However, currently there is a bug caused by the change of look in Google Scholar so all the element names are not the same. In the future, do consider this one.