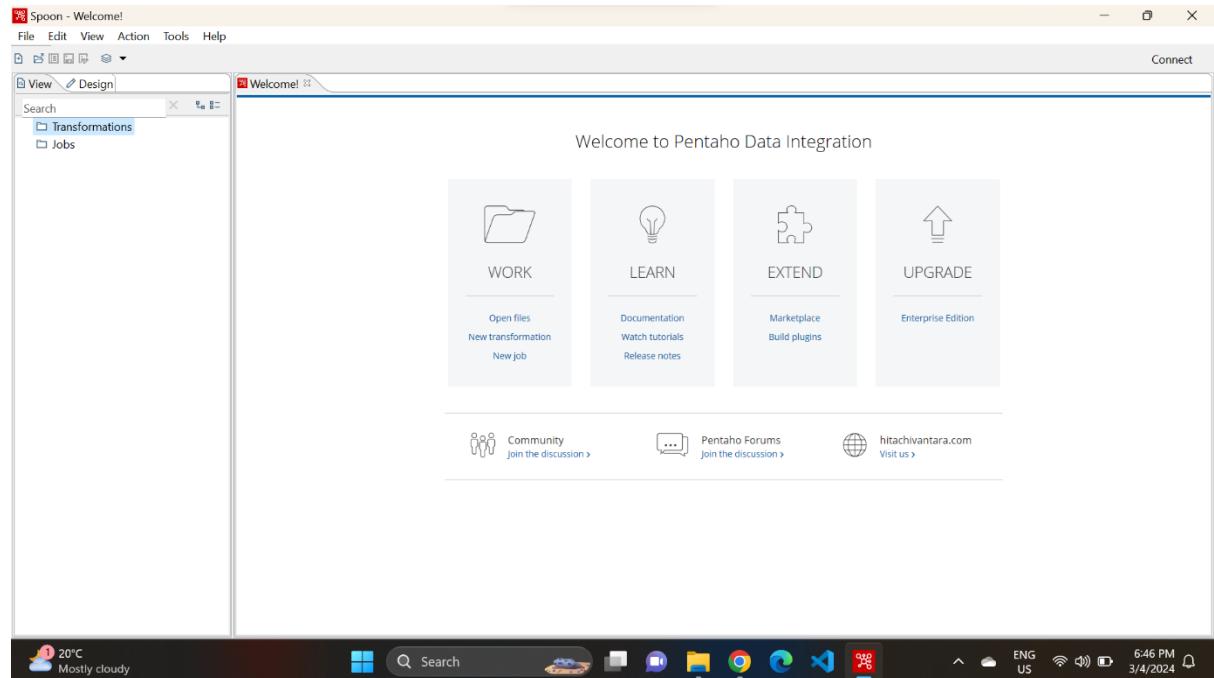


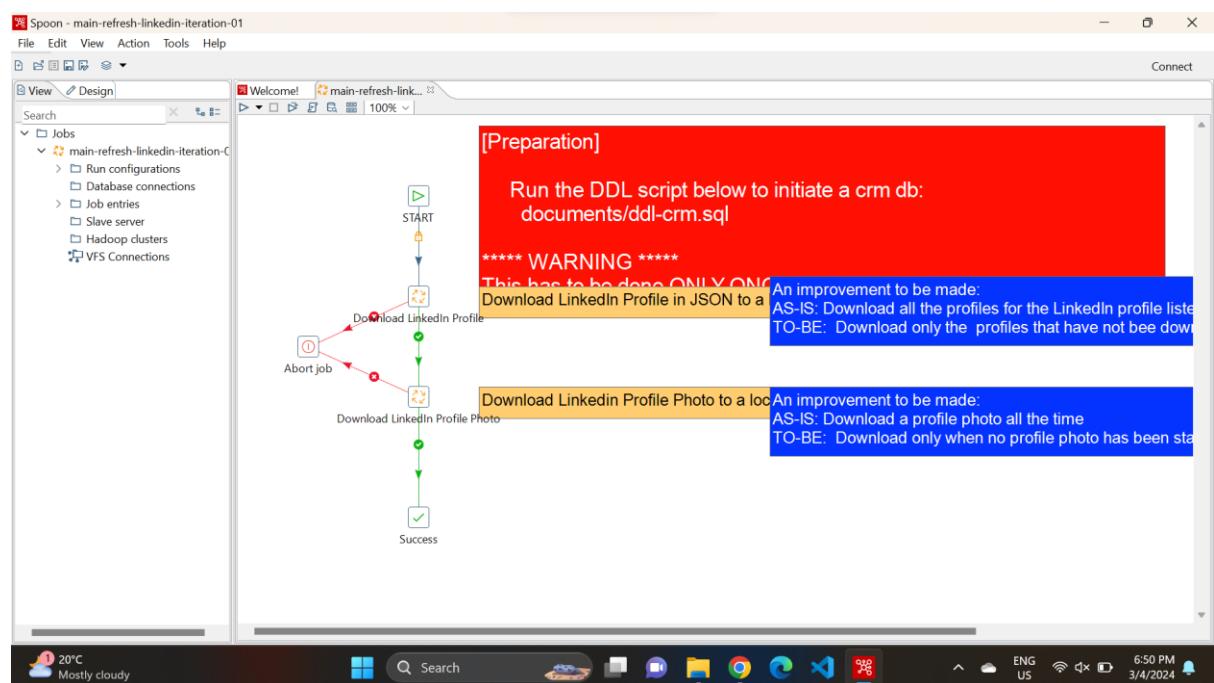
## Project Phase 1- Ingesting LinkedIn Profile

### Deliverable A: Documentation

Step 1: Open the Spoon executable jar file for Pentaho.



Opening the main refresh linkedin iteration kjb file



Checking Proxycurl, before the first API request, there are 10 credits available.

The screenshot shows the Proxycurl API dashboard. On the left, a sidebar lists 'Proxycurl API' sections: API Key, API Logs, Work Email Lookup Logs, Docs, and Addons. Under Addons, it lists 'Sapiengraph - B2B Enrichment in Google Sheets'. The main area displays a message: 'The aha-moment of Proxycurl comes right after you make the first API request. The Person Profile Endpoint takes any "LinkedIn Profile URL" and returns enriched and fresh data on the profile. Do try it on your own profile!' Below this is a button 'See API docs for Person Profile Endpoint'. The 'OVERVIEW' section shows the 'Proxycurl API' heading and tabs for API Key, API Logs, Work Email Lookup Logs. The 'API Key' tab is selected, showing a table for 'Bearer Token (API Key)'. The table has columns: TOKEN, LABEL, and ACTION. One entry is shown: 'kuO2RwiGcI3fSVtGEDD1cg' under 'TOKEN', 'default' under 'LABEL', and a 'Chat with us' button under 'ACTION'. A 'Generate more API keys' button is also present.

Step 2: In SQLYog, we will first create the database crm, then insert the random three linkedin profile id's.

The screenshot shows the SQLYog Community 64 interface. The connection is set to 'mysql80localhost'. The left sidebar shows the database structure with schemas like 'adventureworks', 'information\_schema', 'mysql', 'performance\_schema', and 'sys'. The main query editor window contains the following MySQL code:

```
/* Initialize DB. This must be done ONLY ONCE. It will delete the existing data otherwise */
DROP DATABASE IF EXISTS crm;
CREATE DATABASE crm CHARACTER SET utf8mb4;
USE crm;
DROP TABLE IF EXISTS src_linkedin;
CREATE TABLE src_linkedin (
    linkedin_public_id VARCHAR(100) NOT NULL,
    profile_filename_full VARCHAR(200),
    profile_filename_short VARCHAR(200),
    process_status VARCHAR(50) NOT NULL DEFAULT 'Requested',
    process_datetime TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    PRIMARY KEY(linkedin_public_id)
);
-- insert a sample linkedin profile id
INSERT INTO src_linkedin (linkedin_public_id) VALUES ('mauricedawson');
INSERT INTO src_linkedin (linkedin_public_id) VALUES ('yuhan-ding-2122b588');
INSERT INTO src_linkedin (linkedin_public_id) VALUES ('interlee');

DROP TABLE IF EXISTS profile_section;
CREATE TABLE profile_section (
```

At the bottom, the status bar shows: http://www.webyog.com, Exec: 0 sec, Total: 0 sec, Ln 24, Col 65, Connections: 1, Upgrade to SQLYog Ultimate, ENG US, 6:59 PM, 3/4/2024.

Query is successful.

The screenshot shows the SQLyog Community 64 interface. The title bar indicates the connection is to mysql80localhost/crm as root@localhost\*. The main window displays a query editor with the following SQL code:

```
Format all queries : Reason #61 to upgrade
Query 1 + [+] [x]
CREATE TABLE src_linkedin (
    linkedin_public_id VARCHAR(100) NOT NULL,
    profile_filename_full VARCHAR(200),
    profile_filename_short VARCHAR(200),
    process_status VARCHAR(50) NOT NULL DEFAULT 'Requested',
    process_datetime TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    PRIMARY KEY(linkedin_public_id)
);

-- insert a sample linkedin profile id
INSERT INTO src_linkedin (linkedin_public_id) VALUES ('mauricedawson');
INSERT INTO src_linkedin (linkedin_public_id) VALUES ('yuhan-ding-2122b588');
INSERT INTO src_linkedin (linkedin_public_id) VALUES ('interlee');

DROP TABLE IF EXISTS profile_section;
```

Below the code, the message pane shows:

- 1 Messages
- 2 Table Data
- 3 Info

15 queries executed, 15 success, 0 errors, 5 warnings

Query: DROP DATABASE IF EXISTS CRM

0 row(s) affected, 1 warning(s)

Execution Time : 0.032 sec  
Transfer Time : 0 sec  
Total Time : 0.032 sec

All

Query batch completed successfully

Exec: 0.416 sec Total: 0.425 sec Ln 18, Col 3 Connections: 1 Upgrade to SQLyog Ultimate

System tray icons include: 20°C Mostly cloudy, Search, File Explorer, Google Chrome, Task View, Taskbar icons, ENG US, 6:59 PM, 3/4/2024.

API request is sent.

The screenshot shows the SQLyog Community 64 interface. The title bar indicates the connection is to mysql80localhost/crm as root@localhost\*. The main window displays the results of the previous query in a table format:

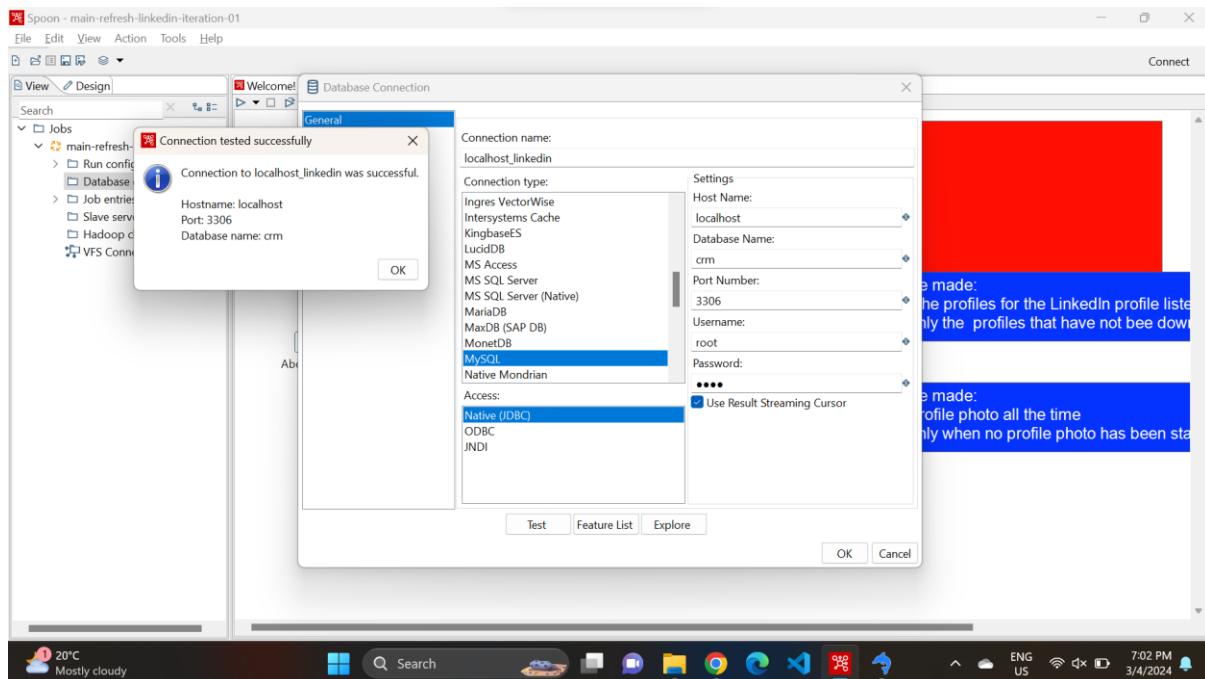
linkedin_public_id	profile_filename_full	profile_filename_short	process_status	process_datetime
interlee	(NULL)	(NULL)	Requested	2024-03-04 18:59:15
mauricedawson	(NULL)	(NULL)	Requested	2024-03-04 18:59:15
yuhan-ding-2122b588	(NULL)	(NULL)	Requested	2024-03-04 18:59:15
*	(NULL)	(NULL)	Requested	CURRENT_TIMESTAMP

Database: CRM Table: src\_linkedin

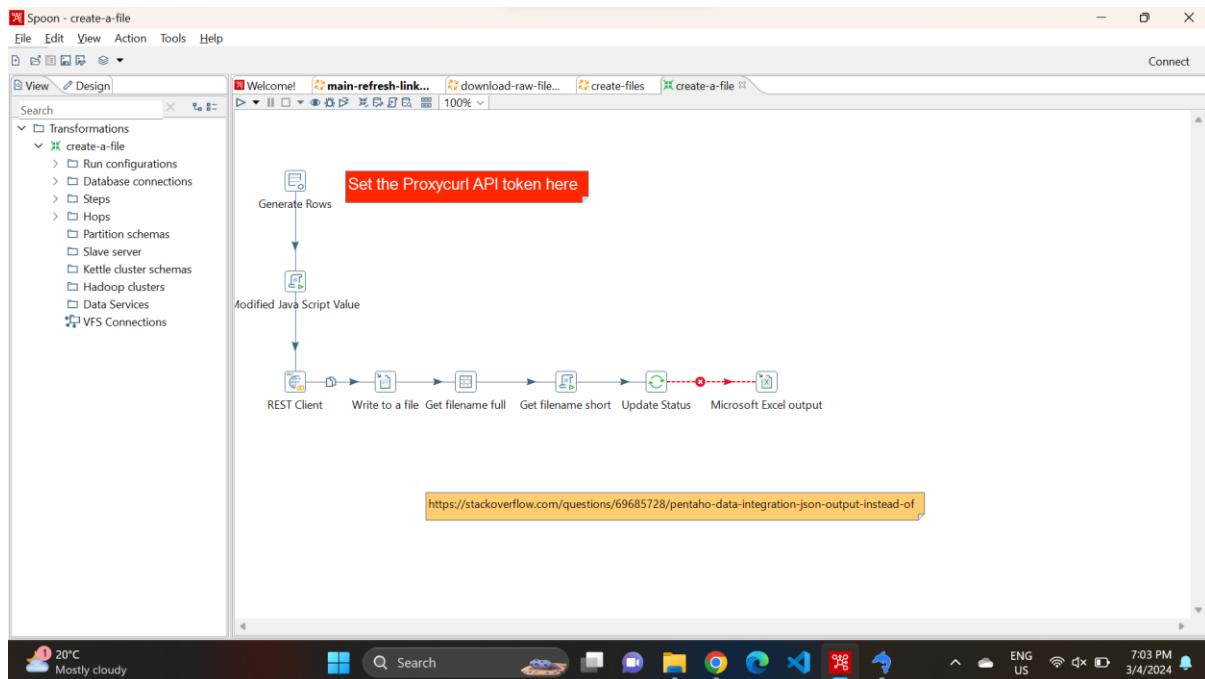
Ready 3 row(s) Connections: 1 Upgrade to SQLyog Ultimate

System tray icons include: 20°C Mostly cloudy, Search, File Explorer, Google Chrome, Task View, Taskbar icons, ENG US, 7:01 PM, 3/4/2024.

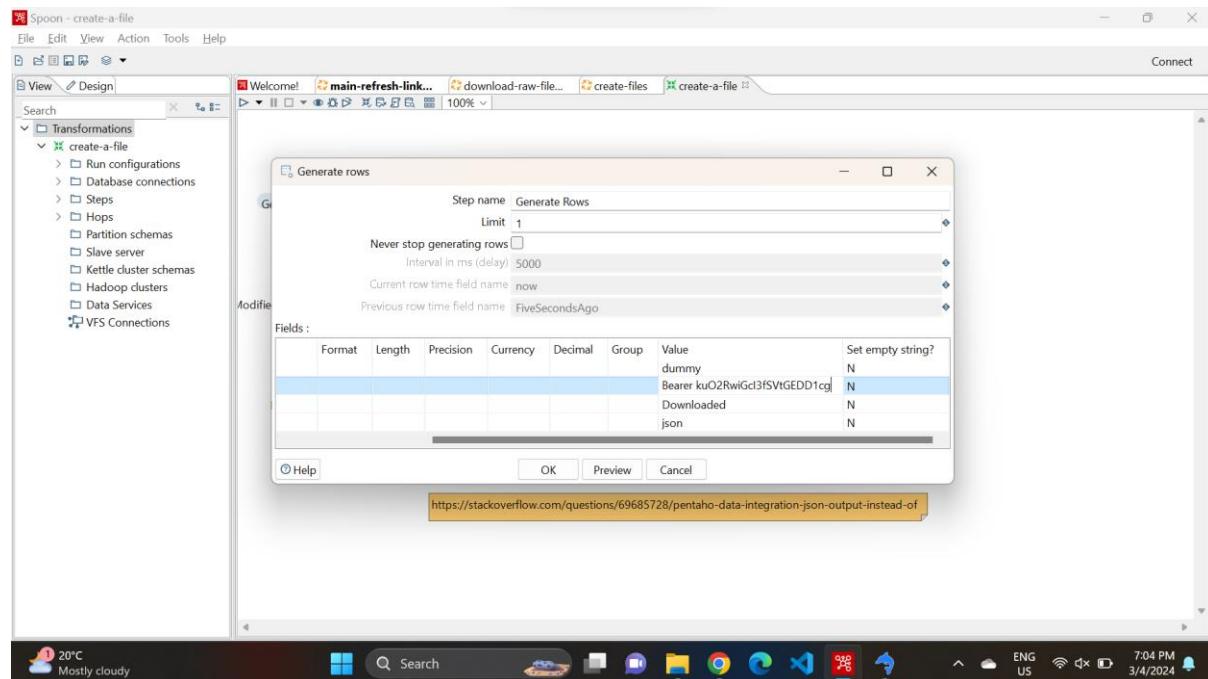
### Step 3: Connecting to the CRM Database in Pentaho.



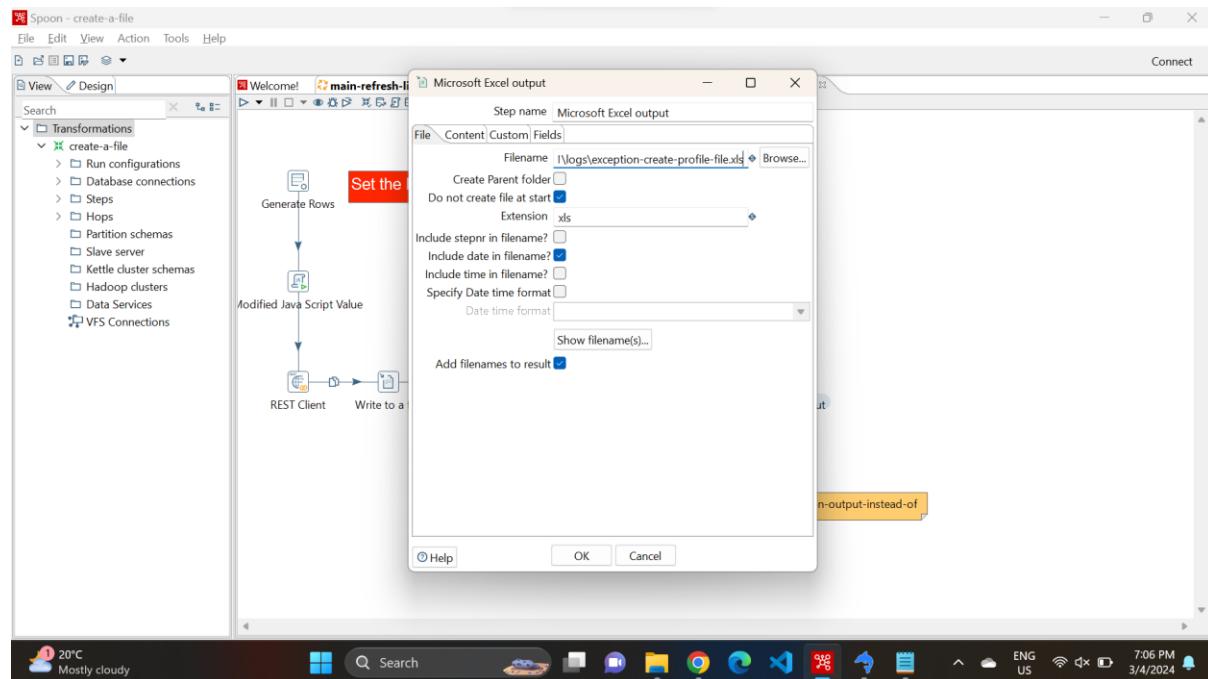
In the download raw files section, we have to set the Proxycurl API key.



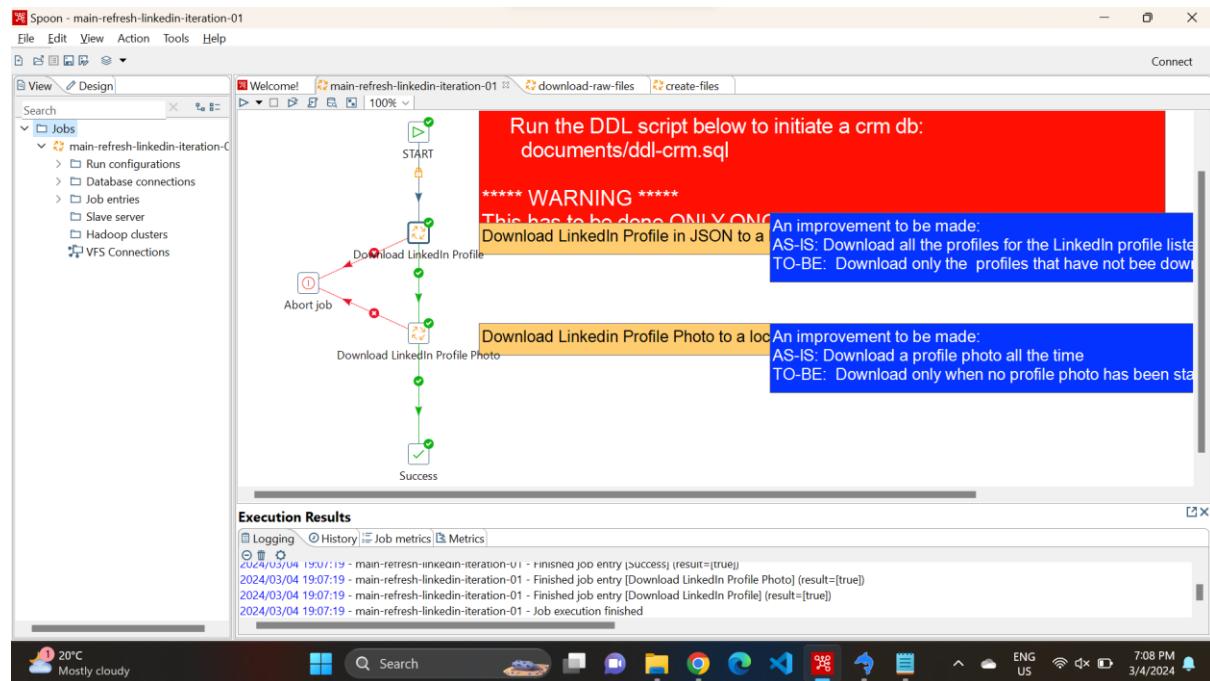
Here we have set the Proxycurl API key in the headerAuth.



Also, we have to set the path for the Microsoft Excel Logs output.

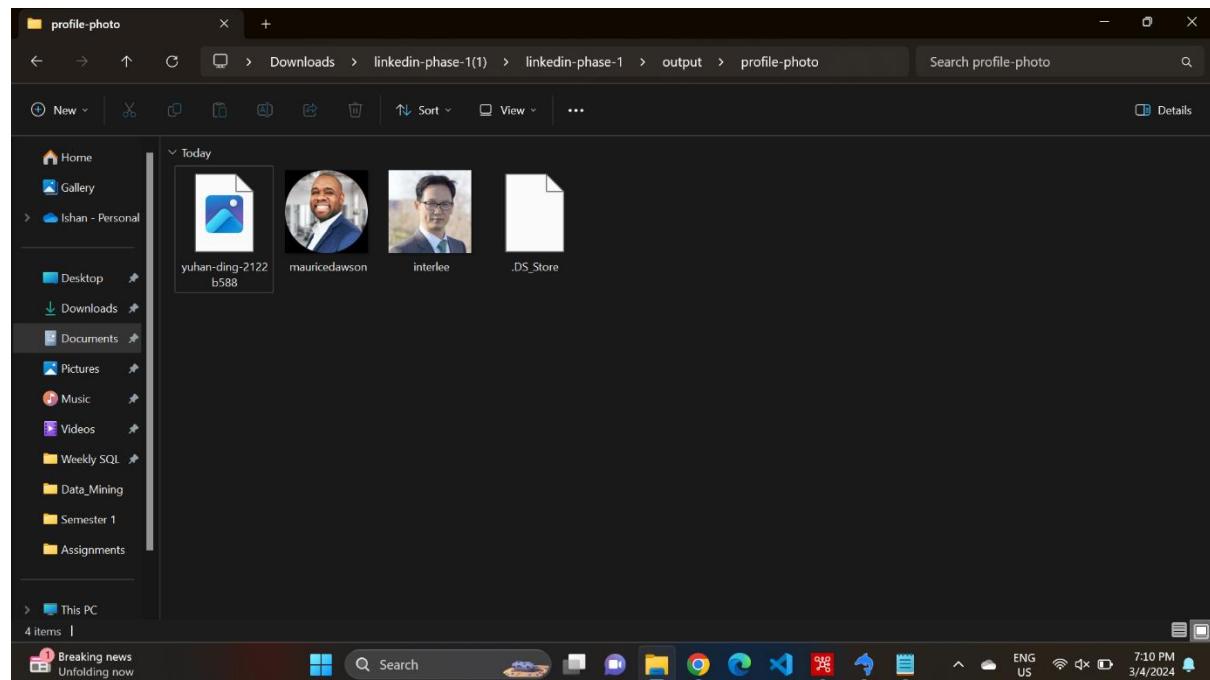


Step 4: After the setup, we will run the main file and observe the execution results.

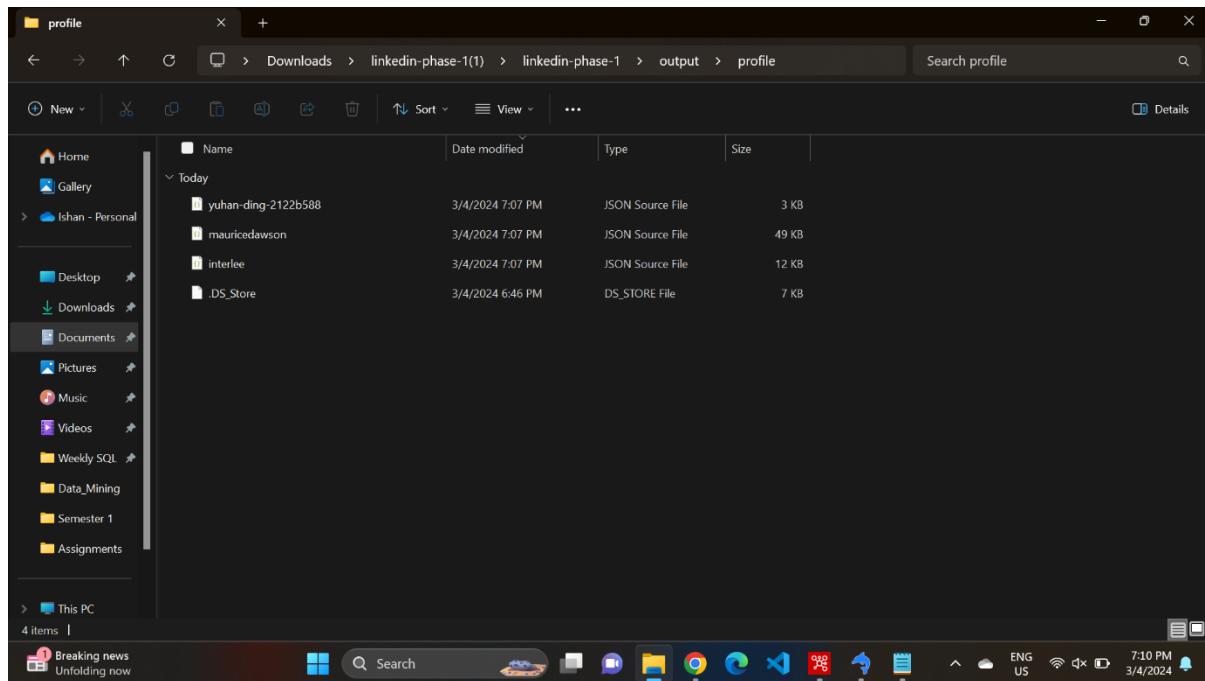


Execution is a success. The profile and profile photos have been downloaded.

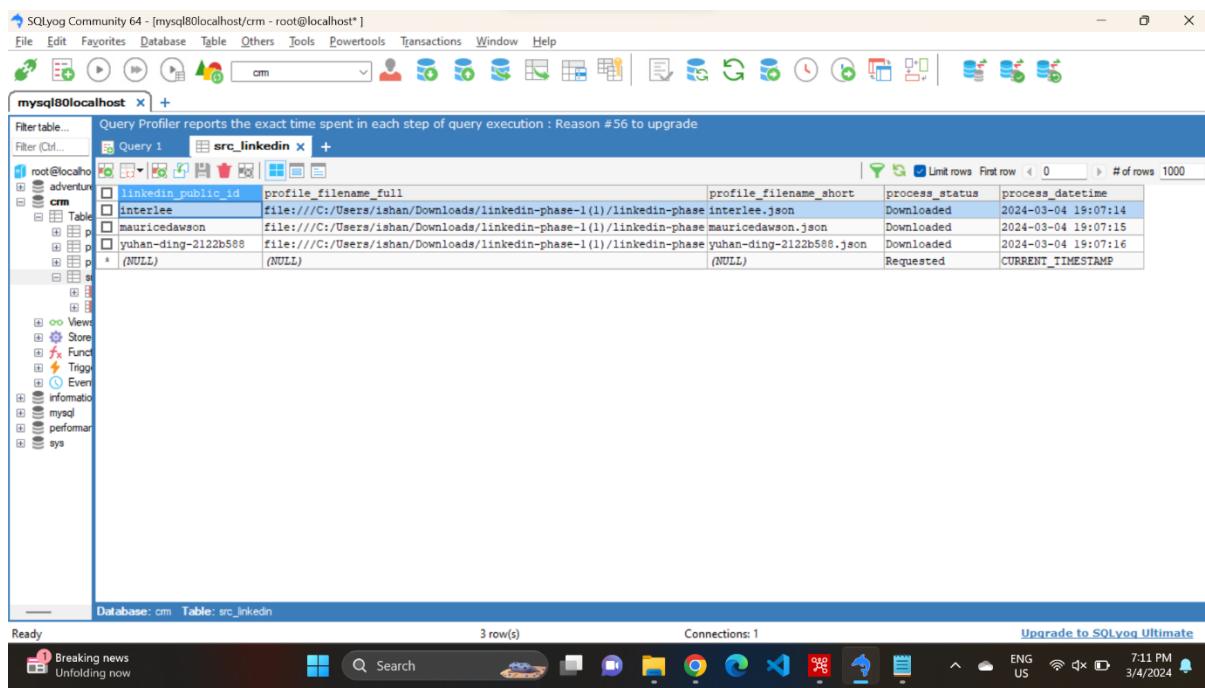
Here we can see the profile photos have been downloaded. One image has an issue with download, so will download another profile as replacement.



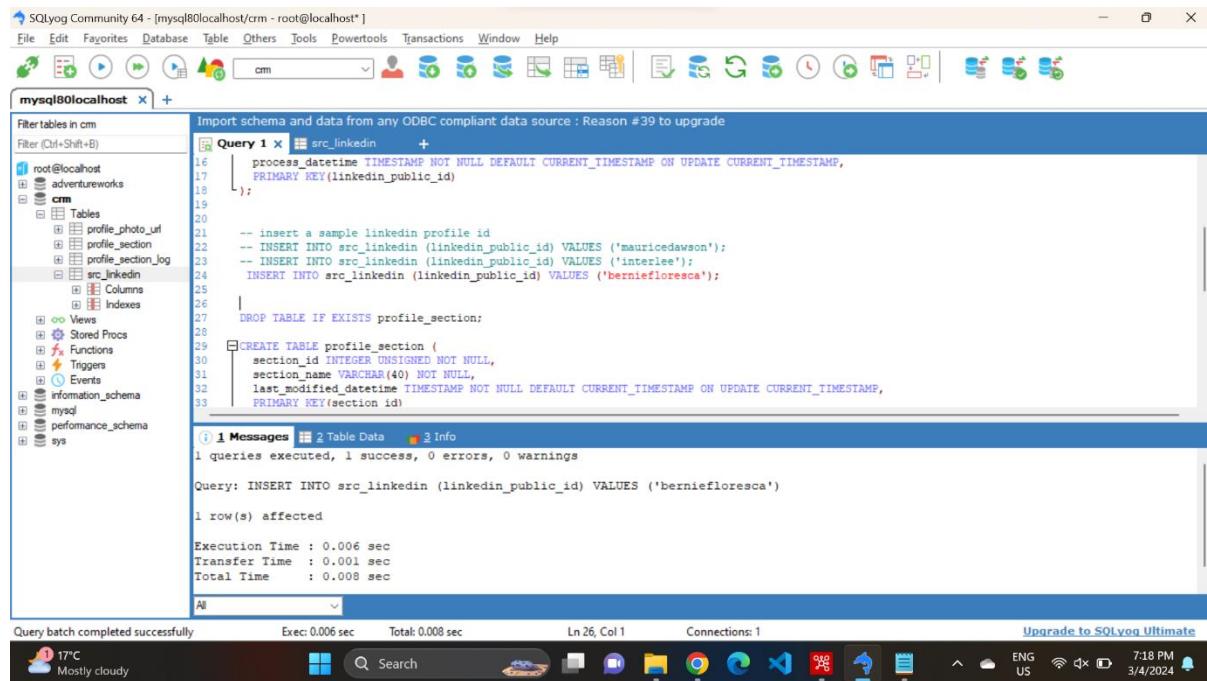
Here are the profiles downloaded in JSON format.



Checking in src\_linkedin file, we can observe that the process is complete and it shows as Downloaded.



Now we will run an alternate profile id to get an extra profile photo and profile json.

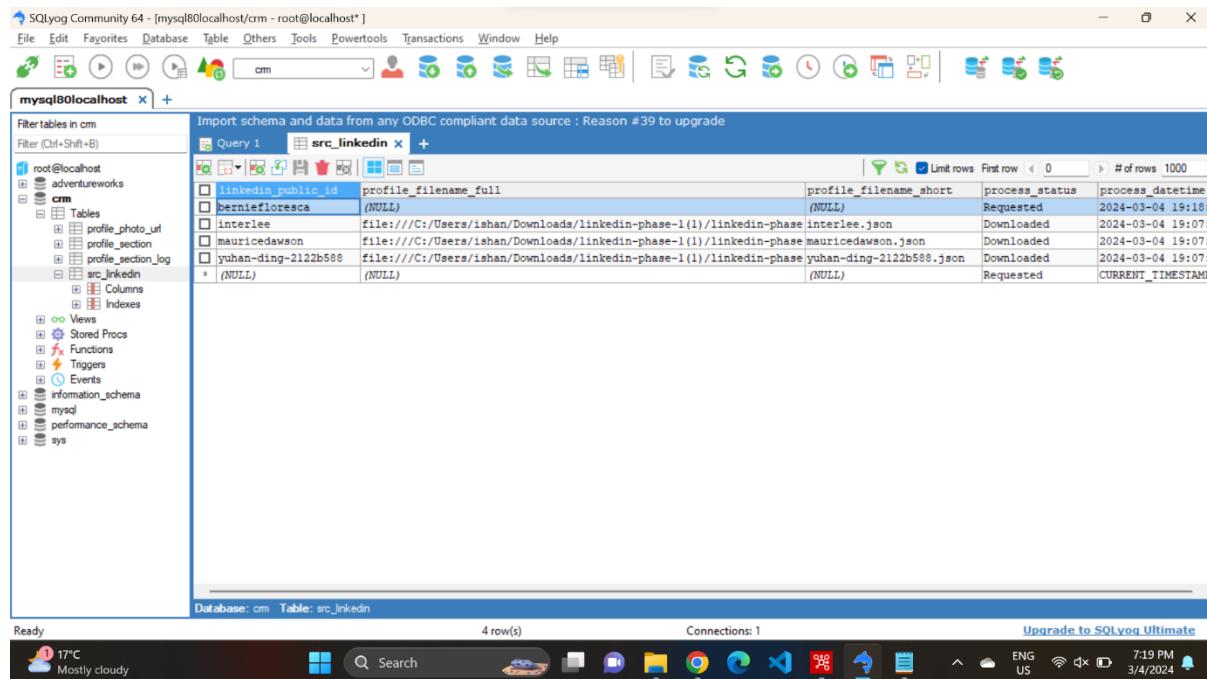


```
Query 1 x src_linkedin +  
16     process_datetime TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
17     PRIMARY KEY(linkedin_public_id)  
18 );  
19  
20  
21 -- insert a sample linkedin profile id  
22 -- INSERT INTO src_linkedin (linkedin_public_id) VALUES ('mauricedawson');  
23 -- INSERT INTO src_linkedin (linkedin_public_id) VALUES ('interlee');  
24 INSERT INTO src_linkedin (linkedin_public_id) VALUES ('berniefloresca');  
25  
26  
27 DROP TABLE IF EXISTS profile_section;  
28  
29 CREATE TABLE profile_section (  
30     section_id INTEGER UNSIGNED NOT NULL,  
31     section_name VARCHAR(40) NOT NULL,  
32     last_modified_datetime TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
33     PRIMARY KEY(section_id)  
1 Messages 2 Table Data 3 Info  
1 queries executed, 1 success, 0 errors, 0 warnings  
Query: INSERT INTO src_linkedin (linkedin_public_id) VALUES ('berniefloresca')  
1 row(s) affected  
Execution Time : 0.006 sec  
Transfer Time : 0.001 sec  
Total Time : 0.008 sec  
All
```

Query batch completed successfully Exec: 0.006 sec Total: 0.008 sec Ln 26, Col 1 Connections: 1 Upgrade to SQLyog Ultimate

17°C Mostly cloudy

Here the id is requested.

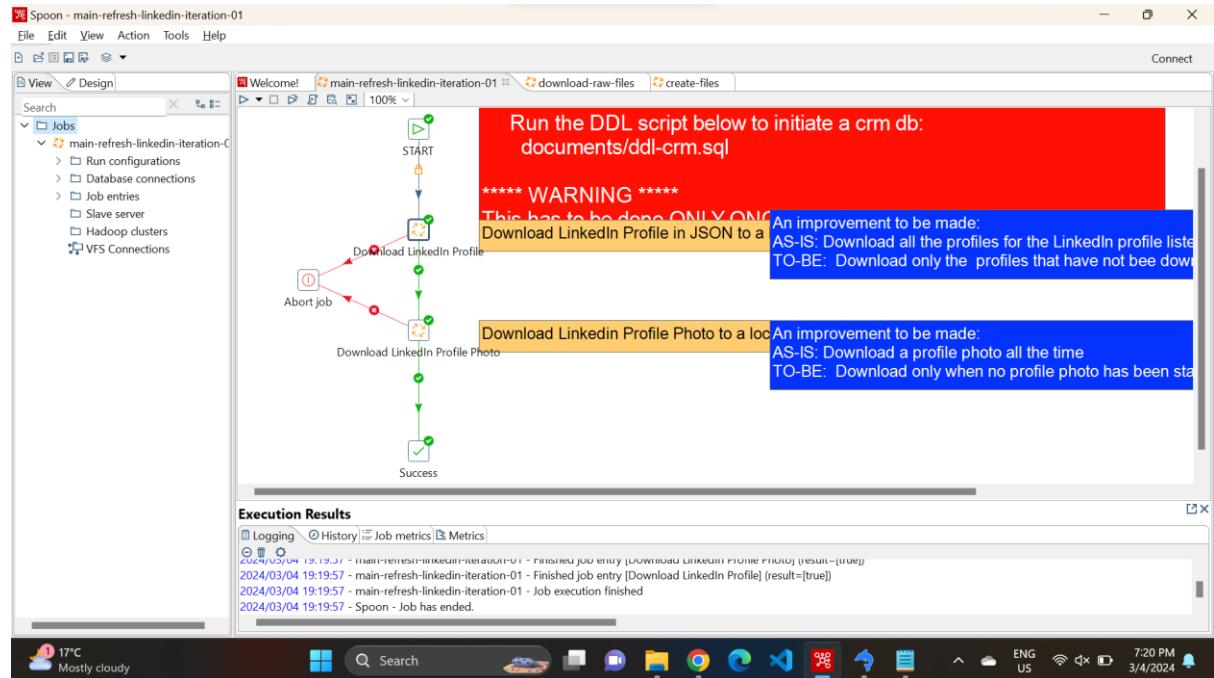


linkedin_public_id	profile_filename_full	profile_filename_short	process_status	process_datetime
berniefloresca	(NULL)	(NULL)	Requested	2024-03-04 19:18:
interlee	file:///C:/Users/ishan/Downloads/linkedin-phase-1(1)/linkedin-phase_interlee.json	interlee.json	Downloaded	2024-03-04 19:07:
mauricedawson	file:///C:/Users/ishan/Downloads/linkedin-phase-1(1)/linkedin-phase_mauricedawson.json	mauricedawson.json	Downloaded	2024-03-04 19:07:
yuhan-ding-2122b588	file:///C:/Users/ishan/Downloads/linkedin-phase-1(1)/linkedin-phase_yuhan-ding-2122b588.json	yuhan-ding-2122b588.json	Downloaded	2024-03-04 19:07:
*	(NULL)	(NULL)	Requested	CURRENT_TIMESTAMP

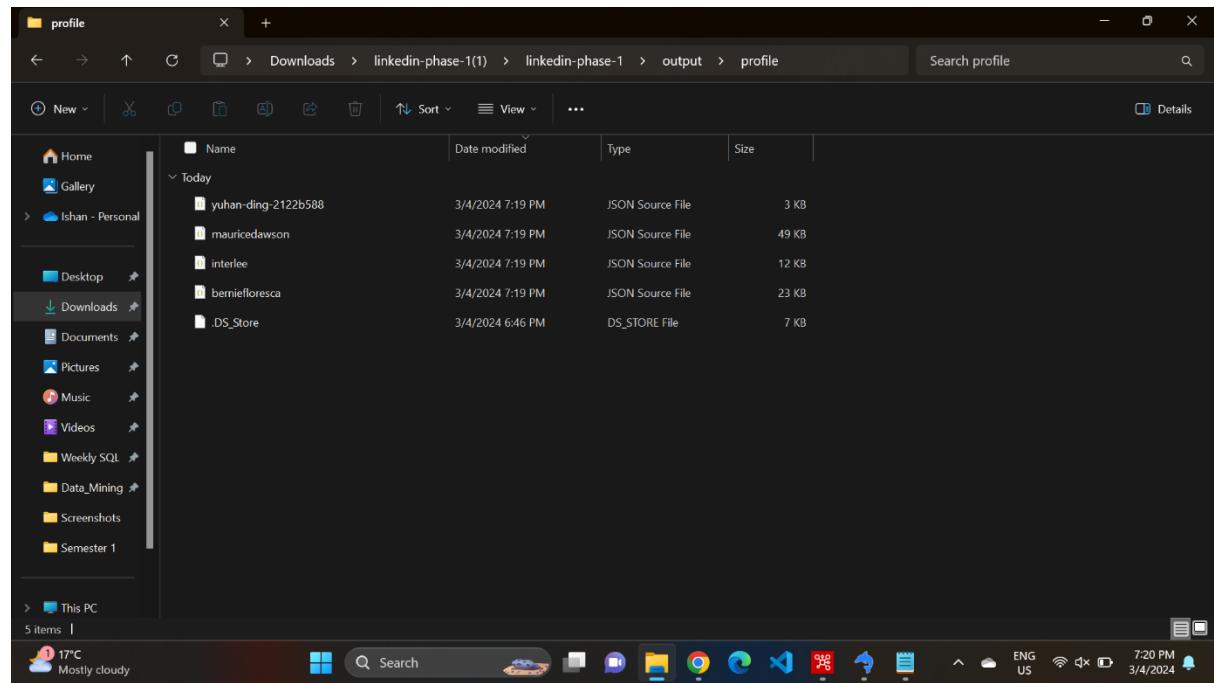
Database: cm Table: src\_linkedin 4 row(s) Connections: 1 Upgrade to SQLyog Ultimate

Ready 17°C Mostly cloudy

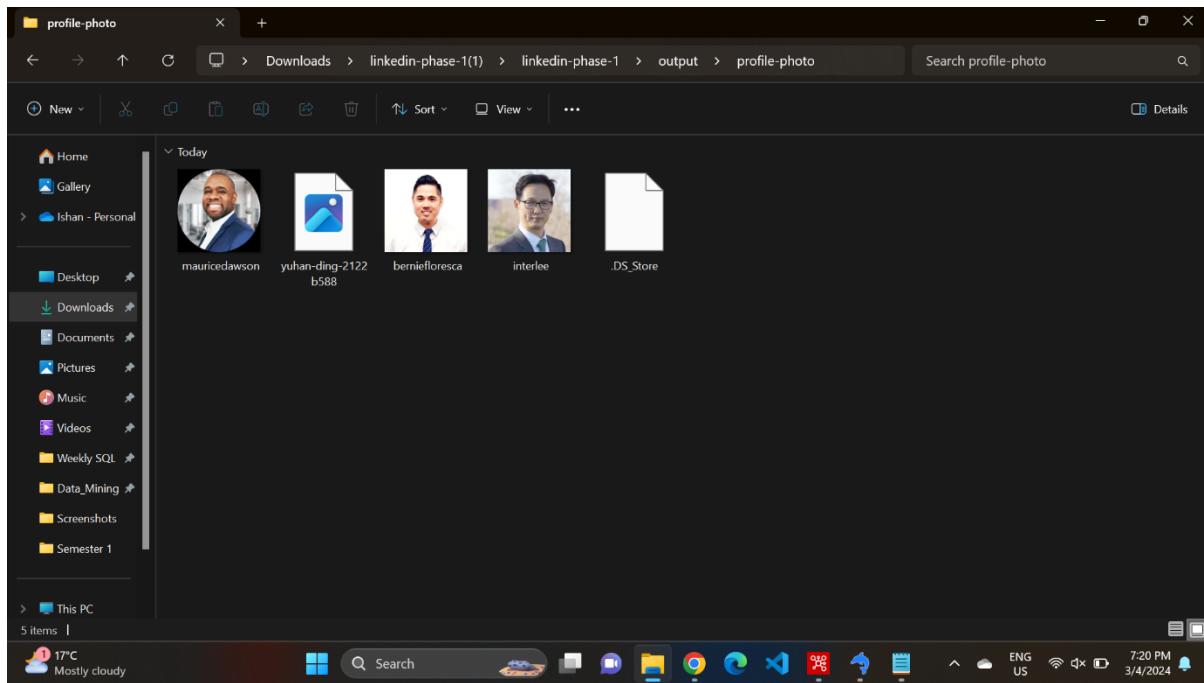
Now, we again run the main file and observe the execution results.



The new profile has been downloaded.



The new profile photo has been downloaded.



The src\_linkedin file also shows that the process is downloaded and completed.

linkedin_public_id	profile_filename_full	profile_filename_short	process_status	process_datetime
berniefloresca	file:///C:/Users/ishan/Downloads/linkedin-phase-1(l)/linkedin-phase berniefloresca.json	berniefloresca.json	Downloaded	2024-03-04 19:19:50
interlee	file:///C:/Users/ishan/Downloads/linkedin-phase-1(l)/linkedin-phase interlee.json	interlee.json	Downloaded	2024-03-04 19:07:14
mauricedawson	file:///C:/Users/ishan/Downloads/linkedin-phase-1(l)/linkedin-phase mauricedawson.json	mauricedawson.json	Downloaded	2024-03-04 19:07:15
yuhan-ding-2122b588	file:///C:/Users/ishan/Downloads/linkedin-phase-1(l)/linkedin-phase yuhan-ding-2122b588.json	(NULL)	Downloaded	2024-03-04 19:07:16
*	(NULL)	(NULL)	Requested	CURRENT_TIMESTAMP

As we can observe, multiple API's were called and now there are no credits remaining. These are the API logs.

The screenshot shows the Proxycurl API dashboard on a Windows desktop. The browser tab is "nubela.co/proxycurl/dashboard/api-logs". The main content area is titled "Proxycurl API" with tabs for "API Key", "API Logs", and "Work Email Lookup Logs". The "API Logs" tab is selected, displaying a table of log entries. The table has columns: TIMESTAMP, CREDIT USED, ENDPOINT, and QUERY STRING. The log entries are:

TIMESTAMP	CREDIT USED	ENDPOINT	QUERY STRING
2024-03-05 01:19:55.308240+00:00	1	/api/v2/linkedin	{"personal_email": "include", "fallback_to_cache": "on"}
2024-03-05 01:19:53.561129+00:00	3	/api/v2/linkedin	{"personal_email": "include", "fallback_to_cache": "on"}
2024-03-05 01:19:51.563211+00:00	3	/api/v2/linkedin	{"personal_email": "include", "fallback_to_cache": "on"}
2024-03-05 01:19:50.423234+00:00	2	/api/v2/linkedin	{"personal_email": "include", "fallback_to_cache": "on"}
2024-03-05 01:07:17.020444+00:00	1	/api/v2/linkedin	{"personal_email": "include", "fallback_to_cache": "on"}

The sidebar on the left includes sections for "Proxycurl API" (with "API Key", "API Logs", "Work Email Lookup Logs", and "Docs"), "Addons" (with "SapienGraph - B2B Enrichment in Google Sheets"), and a "Buy credits" button. The Windows taskbar at the bottom shows the date (3/4/2024), time (7:22 PM), battery level (ENG US), and weather (17°C Mostly cloudy).