Extract, transform and load

The purpose of this task is to introduce you to the basic ideas of "extract, transform and load" or ETL, one of the basic tasks of a data scientist.

Before doing this task, it may be helpful for you to install Jupyter Notebook on your local system. The easiest way to do this is to install Anaconda, which comes with all the packages to do this task.

Reading

Before trying the task, read about the following topics. Software engineers often have to use different software packages. Read and try the tutorials or documentation to figure out how to do things.

API

- What is an API request?
- What's the difference between a GET and POST request?
- How do you make an API request in Python, and how do you get the result? (Hint: there is a Python package, `requests`.)

JSON

- The output of an API is nearly always in JSON format.
- You can think of JSON as similar to a nested dict in Python (a dict inside a dict).
- There is a Python library, `json`, that allows you to convert a JSON text string into a Python dictionary. Check the tutorial and play around with it for a while.

Pandas

- Try a simple tutorial for the Pandas package https://pandas.pydata.org/.
- What is a dataframe?
- How do you create a dataframe in Pandas?
    - How can you convert JSON data to a dataframe? How about a dictionary?
- How can you convert a data structure such as a dictionary into a dataframe?

Task

In this task we will carry out a simple data mining exercise. This can be done in about 10-20 lines of Python code.

1. Make a GET request on these two addresses. What is the output? The only difference between the two links is that the `cryptoID` field is different.

https://api.coinmarketcap.com/data-api/v3/token-unlock/event?type=past&cryptoId=5690&page=1&limit=10&enableSmallUnlocks=true
https://api.coinmarketcap.com/data-api/v3/token-unlock/event?type=past&cryptoId=6719&page=1&limit=10&enableSmallUnlocks=true

2. Take the GET response of each API call and convert it into a dictionary. Can you find the `data->tokenEvent` field?

3. Extract the information from the `data->tokenEvent` field and load it into a Pandas dataframe. What does the data structure look like?

4. Create a combined dataframe with the data from id=5690 and 6719. In the final dataframe, you should be able to mark down which rows of data belong to 5690 and which belong to 6719, as well as the time that you did the extraction.

Sample result

	amount	time	percentage	allocations	crypto_id	etl_timestamp
0	5.965887e+05	2024-10-01T00:00:00.000Z	0.11	[{'allocationName': 'Inflation', 'amount': 596...	5690	2024-10-11 13:42:05.112640
1	6.026148e+05	2024-09-01T00:00:00.000Z	0.11	[{'allocationName': 'Inflation', 'amount': 602...	5690	2024-10-11 13:42:05.112640
2	6.087018e+05	2024-08-01T00:00:00.000Z	0.11	[{'allocationName': 'Inflation', 'amount': 608...	5690	2024-10-11 13:42:05.112640
3	6.148503e+05	2024-07-01T00:00:00.000Z	0.12	[{'allocationName': 'Inflation', 'amount': 614...	5690	2024-10-11 13:42:05.112640
4	6.210609e+05	2024-06-01T00:00:00.000Z	0.12	[{'allocationName': 'Inflation', 'amount': 621...	5690	2024-10-11 13:42:05.112640
5	6.273343e+05	2024-05-01T00:00:00.000Z	0.12	[{'allocationName': 'Inflation', 'amount': 627...	5690	2024-10-11 13:42:05.112640
6	6.336710e+05	2024-04-01T00:00:00.000Z	0.12	[{'allocationName': 'Inflation', 'amount': 633...	5690	2024-10-11 13:42:05.112640
7	6.400717e+05	2024-03-01T00:00:00.000Z	0.12	[{'allocationName': 'Inflation', 'amount': 640...	5690	2024-10-11 13:42:05.112640
8	6.465371e+05	2024-02-01T00:00:00.000Z	0.12	[{'allocationName': 'Inflation', 'amount': 646...	5690	2024-10-11 13:42:05.112640
9	6.530678e+05	2024-01-01T00:00:00.000Z	0.12	[{'allocationName': 'Inflation', 'amount': 653...	5690	2024-10-11 13:42:05.112640
0	8.732900e+07	2024-09-17T00:00:00.000Z	0.81	[{'allocationName': 'Early Team & Advisors', '...	6719	2024-10-11 13:42:06.114337
1	7.144700e+07	2024-08-17T00:00:00.000Z	0.66	[{'allocationName': 'Early Team & Advisors', '...	6719	2024-10-11 13:42:06.114337
2	7.144700e+07	2024-07-17T00:00:00.000Z	0.66	[{'allocationName': 'Foundation', 'amount': 12...	6719	2024-10-11 13:42:06.114337
3	8.732900e+07	2024-06-17T00:00:00.000Z	0.81	[{'allocationName': 'Edge & Node', 'amount': 1...	6719	2024-10-11 13:42:06.114337