

# PARSER DEVELOPMENT AGENDA – Wednesday, February 12, 2025

## Table of Contents

- GENERAL REQUESTS .....2**
- HARI.....4**
  - United Kingdom .....4
  - Ireland .....4
  - NorthernIreland .....4
- RAHUL.....5**
  - NICARAGUA .....5
  - NEW ZEALAND .....5
- RIKTESH.....6**
  - PANAMA.....6
  - PERU.....6
  - TAIWAN.....7
  - SINGAPORE.....8
- NEELAKSH .....8**
  - AUSTRALIA.....9

## GENERAL REQUESTS

Here is the desired workflow we want in our product. We will maintain a spreadsheet of URLs for every region, and then a python function will periodically go and fetch data from the URL and store in a consistent DB format.



TO ACHIEVE THIS ---

Please write a function for every country or zone (assigned to you) like:

```
import json

def REGION_NAME(url: str) -> str:
    """
    Fetch energy data from the provided URL and return a JSON string with
    energy metrics.
```

Note:

This is a dummy function meant to show how you might structure your code.

Please perform an HTTP GET (or POST) as mentioned in the region table to extract the data, process data from the URL.

The returned JSON contains the following keys, each representing an energy metric:

- BAT (float): battery storage systems.
- COL (float): coal-based production.
- GEO (float): geothermal energy production.
- NG (float): natural gas energy production.
- NUC (float): nuclear power generation.
- OES (float): other energy sources not categorized elsewhere.
- OIL (float): oil production.
- OTH (float): miscellaneous or other energy types.
- PS (float): pumped storage systems.
- SNB (float): a specific subset of solar energy (dummy field; may represent solar energy from a new baseline technology).
- SUN (float): standard solar (sunlight) energy production.
- UES (float): unspecified energy sources.
- WAT (float): water or hydroelectric energy production.
- WND (float): wind energy production.

Args:

url (str): The URL endpoint from which to (hypothetically) fetch energy data.

Returns:

str: A JSON string containing energy data metrics.  
"""

# Dummy data values for demonstration purposes.

Note that if the country you are fetching has 24 datapoints ... the returned json should contain 24 sets of data.

```
energy_data = {
    "timestamp": YYYY-MM-DD HH:00:00,
    "updatedAt": currentTime stamp, in YYYY-MM-DD HH:mm:ss format
    "BAT": 1.0,
    "COL": 2.0,
    "GEO": 3.0,
    "NG": 4.0,
    "NUC": 5.0,
    "OES": 6.0,
    "OIL": 7.0,
    "OTH": 8.0,
    "PS": 9.0,
    "SNB": 10.0,
    "SUN": 11.0,
    "UES": 12.0,
    "WAT": 13.0,
    "WND": 14.0,
}

# Convert the dictionary to a JSON string and return it.
return json.dumps(energy_data)
```

THEN →

1. Test the function.
2. Save it as `regionName.py`
3. Keep notes of which data is daily, hourly, half-hourly, etc. We will add estimation logic accordingly if needed.
4. Note that if the country you are fetching has 24 datapoints ... the returned json should contain 24 sets of data.
5. If 10, 15, or 30 min data is available from the API, take only the data at the hourly values.. no averaging or anything is needed over the hour.
6. For whatever value is not found, we will write it as 0 in the database.. make sure all values are 0 or something in the returned JSON.

7. Inside the function, you can use some electricitymaps parser code .. but don't use any code that requires us to install the electricitymaps package into our system.
8. Note the units.. our intended units in the database is MW .. if some API is returning as GW .. note it and let us know.
9. We are not considering "Imports" during this phase of our implementation.
10. The highlighted portions you might see in the URL is an example value, and needs to be input accordingly by you in the code.

## HARI

### United Kingdom

GET Request URL	Things to do here
<a href="https://api.carbonintensity.org.uk/generation/2025-02-10T12:00Z/pt24h">https://api.carbonintensity.org.uk/generation/2025-02-10T12:00Z/pt24h</a>	<p>Run this once daily. 30 minute daily interval data is available. Extract only hourly values.</p> <p>Only the highlighted value of date time input is to be changed.</p> <p>Note these are only percentages .. as of now multiply</p>

### Ireland

GET Request URL	Things to do here
<a href="https://www.eirgrid.ie/api/graph-data?area=fuelmix&amp;region=ROI&amp;date=12%20Feb%202025">https://www.eirgrid.ie/api/graph-data?area=fuelmix&amp;region=ROI&amp;date=12%20Feb%202025</a>	<p>This data updates every 15 minutes.. so we will have to run this request hourly.</p> <p>read the English names and map it to our JSON format.</p>

### NorthernIreland

GET Request URL	Things to do here
<a href="https://www.eirgrid.ie/api/graph-data?area=fuelmix&amp;region=NI&amp;date=12%20Feb%202025">https://www.eirgrid.ie/api/graph-data?area=fuelmix&amp;region=NI&amp;date=12%20Feb%202025</a>	<< same way as Ireland above >>

## RAHUL

### NICARAGUA

GET Request URL	Things to here
<a href="https://www.cndc.org.ni/graficos/consultarGeneracionPorTipo">https://www.cndc.org.ni/graficos/consultarGeneracionPorTipo</a>	<p>It simply gives the last 24 hours of different energy production metrics. Consider only the “GeneracionTipo” of the response.</p> <p>TERMICA → Thermal (Fossil) HIDROELECTRICA → Hydroelectric GEOTERMICA → Geothermal BIOMASA → Biomass EOLICA → Wind SOLAR → Solar</p> <p>Note that MER → Imports ... check with the Electricity Maps to see how they deal with it? Else, put into into “Others” or “Unknown” category</p>

### NEW ZEALAND

GET Request URL	Things to do here
<a href="https://www.transpower.co.nz/em6/data/current_generation/{{UNIX_EPOCH_TIME_IN_SECONDS}}">https://www.transpower.co.nz/em6/data/current_generation/{{UNIX_EPOCH_TIME_IN_SECONDS}}</a>	1) When recording to database, convert the unix epoch timestamp to the YYYY-MM-DD format

## RIKTESH

### PANAMA

GET Request URL	Things to do here
<a href="https://sitr.cnd.com.pa/m/pub/data/gen.js">https://sitr.cnd.com.pa/m/pub/data/gen.js</a> on?{{UNIX_EPOCH_TIME_IN_SECONDS}}	<ol style="list-style-type: none"><li>1) Note you have to pass the millisecond in the GET request</li><li>2) When recording to database, convert the unix epoch timestamp to the YYYY-MM-DD format</li><li>3) Only get the the values in “pie” and “pie2” ...</li><li>4) Use translator for words like Hidirica (hydro), Termica (Coal), Wind (Eolica) .. save in the db schema given</li></ol>

### PERU

POST Request URL	Things to do here
<a href="https://www.coes.org.pe/Portal/portalinformacion/generacion?fechaInicial=11%2F02%2F2025&amp;fechaFinal=12%2F02%2F2025&amp;indicador=0">https://www.coes.org.pe/Portal/portalinformacion/generacion?fechaInicial=11%2F02%2F2025&amp;fechaFinal=12%2F02%2F2025&amp;indicador=0</a>	<p>Note, this is a post request.</p> <p>Let's collect one day data at a time. Provide initial and final dates in the API call in the format like DD%2F02%MM%2F02%2FYYYY ... make sure indicador=0</p> <p>Then, you will get a very long JSON</p>

	<p>response.</p> <p>Around 22K line, you will get something called “GraficoTipoCombustible” ... we are interested in that.</p> <p>Use this name translation:</p> <p>"DIESEL": "oil",  "RESIDUAL":  "biomass",  "CARBÓN": "coal",  "GAS": "gas",  "HÍDRICO":  "hydro",  " BIOGÁS":  "biomass",  " BAGAZO":  "biomass",  "SOLAR": "solar",  " EÓLICA ": "wind",</p>
--	---

## TAIWAN

GET request URL	Things to do here
<a href="https://www.taipower.com.tw/d006/loadGraph/loadGraph/data/loadfueltype.csv">https://www.taipower.com.tw/d006/loadGraph/loadGraph/data/loadfueltype.csv</a>	<ul style="list-style-type: none"> <li>- This updates every 10 minutes</li> <li>- Directly capture this data into a</li> </ul>

	<p>pandas dataframe</p> <ul style="list-style-type: none"><li>- Whatever value is in the CSV, needs to be multiplied by 10.</li></ul>
--	---

The header column name of the CSV file is not written .. follow this from top to bottom .. whatever name is given in this screenshot:



## SINGAPORE

GET Request URL	Things to do here
<a href="https://www.emcsg.com/ChartServer/blue/ticker">https://www.emcsg.com/ChartServer/blue/ticker</a>	<ul style="list-style-type: none"><li>- Note this this updated once daily</li><li>- Note these values from the JSON response</li><li>- , "Period": "18", "Date": "12 Feb 2025", "</li><li>- { "Label": "Demand", "Value": "6,873MW" }</li><li>- { "Name": "Generator Type Share", "SectionData": [{ "Label": "CCGT/COGEN/TRIGEN", "Value": "97.66% " }, { "Label": "ST", "Value": "1.82% " }, { "Label": "GT", "Value": "0.00% " } ] }</li><li>- Put CCGT/COGEN/TRIGEN as Gas and ST as “Unknown”</li></ul>

## NEELAKSH



## AUSTRALIA

Since you already know the Australia URLs, I am not putting them here, but please follow the function template described earlier for your outputs

For Rottenest Island, ignore the zone for now.