

Documentation of the Data Collection Assessment

1.1 Summary

The file “main.py” contains the entry point to the Python code for collecting data from the NOAA online resources. Currently, this project is able to fetch the Real Time Solar Wind data from the NOAA website, store it in a SQL database table, and update that table on a regularly scheduled interval. The program expects the user to supply a desired action (build, update, or print) and a database table name. A breakdown of the individual files is provided below.

Requirements: This code require the end user to have Python 3 install on their system. It is intended to be ran the the command prompt (see example below).

Example usage for the RTSW table in Windows command prompt:

- Build: “python main.py -b RTSW”
- Update: “python main.py -u RTSW”
- Print: “python main.py -p RTSW”

Example output:

```
C:\VERT-assessments\Data_collection>python main.py -b RTSW
RTSW Data inserted into the database.

C:\VERT-assessments\Data_collection>python main.py -u RTSW
RTSW table already up to date.
Updated table with 0 new entries.
Updated table with 2 new entries.

Keyboard Interrupt.
Manually exiting program...

C:\VERT-assessments\Data_collection>python main.py -p RTSW
('2024-12-23 13:43:00.000', '555.6', '1.67', '169073', '2.89', '-3.93', '2.76', '5.60', None, None, None, '2024-12-23 14:21:15.000')
('2024-12-23 13:44:00.000', '558.2', '1.64', '176760', '3.03', '-3.14', '3.65', '5.69', None, None, None, '2024-12-23 14:22:05.000')
('2024-12-23 13:45:00.000', '559.6', '1.67', '158325', '2.81', '-4.61', '2.44', '5.92', None, None, None, '2024-12-23 14:22:59.000')
('2024-12-23 13:46:00.000', '556.0', '1.69', '162868', '2.89', '-4.52', '2.47', '5.91', None, None, None, '2024-12-23 14:24:14.000')
('2024-12-23 13:47:00.000', '553.8', '1.72', '174936', '3.00', '-4.46', '2.37', '5.87', None, None, None, '2024-12-23 14:25:23.000')
('2024-12-23 13:48:00.000', '553.6', '1.70', '176261', '3.04', '-4.81', '1.50', '5.88', None, None, None, '2024-12-23 14:26:24.000')
('2024-12-23 13:49:00.000', '554.9', '1.75', '168403', '2.93', '-5.24', '0.09', '6.00', None, None, None, '2024-12-23 14:27:18.000')
('2024-12-23 13:50:00.000', '558.2', '1.84', '163120', '2.68', '-5.45', '0.30', '6.08', None, None, None, '2024-12-23 14:28:05.000')
('2024-12-23 13:51:00.000', '558.8', '1.70', '171430', '2.62', '-5.23', '1.50', '6.04', None, None, None, '2024-12-23 14:29:02.000')
('2024-12-23 13:52:00.000', '560.1', '1.86', '163630', '2.51', '-5.26', '1.61', '6.04', None, None, None, '2024-12-23 14:29:57.000')
('2024-12-23 13:53:00.000', '562.3', '1.79', '174504', None, None, None, None, None, None, None, '2024-12-23 14:30:48.000')
('2024-12-23 13:54:00.000', '563.1', '1.79', '176360', '2.21', '-5.49', '0.97', '6.00', None, None, None, '2024-12-23 14:31:45.000')
('2024-12-23 13:55:00.000', '575.1', '1.72', '202339', '1.06', '-6.34', '0.16', '6.43', None, None, None, '2024-12-23 14:31:58.000')
('2024-12-23 13:56:00.000', '573.5', '1.80', '170288', '1.85', '-5.76', '1.25', '6.18', None, None, None, '2024-12-23 14:33:04.000')
('2024-12-23 13:57:00.000', '562.2', '1.85', '165528', '2.25', '-5.79', '1.16', '6.32', None, None, None, '2024-12-23 14:34:48.000')
('2024-12-23 13:58:00.000', '569.7', '2.02', '174191', '2.37', '-5.34', '-0.96', '5.92', None, None, None, '2024-12-23 14:35:19.000')
('2024-12-23 13:59:00.000', '558.6', '1.75', '161391', '2.85', '-4.88', '-2.35', '6.12', None, None, None, '2024-12-23 14:37:03.000')
('2024-12-23 14:00:00.000', '555.7', '1.57', '154758', '2.68', '-4.79', '-2.53', '6.05', None, None, None, '2024-12-23 14:38:15.000')
('2024-12-23 14:01:00.000', '553.3', '1.44', '142081', '2.67', '-4.52', '-3.11', '6.10', None, None, None, '2024-12-23 14:39:25.000')
('2024-12-23 14:02:00.000', '570.6', '1.78', '270204', '0.75', '-4.98', '-3.06', '5.89', None, None, None, '2024-12-23 14:39:15.000')
('2024-12-23 14:03:00.000', '583.0', '2.05', '359470', '-0.19', '-5.44', '-3.16', '6.29', None, None, None, '2024-12-23 14:39:27.000')
('2024-12-23 14:06:00.000', '601.4', '1.48', '282459', None, None, None, None, None, None, None, '2024-12-23 14:41:21.000')
('2024-12-23 14:07:00.000', '600.0', '1.65', '326419', '-0.35', '-5.90', '-2.02', '6.25', None, None, None, '2024-12-23 14:42:26.000')
('2024-12-23 14:08:00.000', '580.2', '1.49', '152815', None, None, None, None, None, None, None, '2024-12-23 14:44:38.000')
```

2.1 Program Files

2.1.1 main.py

This Python file provides the main entry point from the command prompt into the program. It takes in the desired action (build, update, or print) and the database table name. When building, it will first delete previous data stored in the table (if it exists) then create a fresh instance of the named table. When updating, it will fetch new data from the source URL on a predetermined schedule (currently 60 seconds), then add any new data into the named table. Only unique data is retained and the table is sorted according to the time tag.

2.1.2 __init__.py

This file simply defines the Python files as a project for the main file.

2.1.3 config.py

This file contains the variable to define the output database name, the update schedule interval, and any NOAA source URLs. Currently only the RTSW URL exists, though more can be added.

2.1.4 fetcher.py

This file contains a single function to fetch the data from a NOAA source provided a URL. It attempts to collect and parse that data in a JSON format. If successful, it returns an error code of "0" and the parsed data. There are a number of common errors that are assigned their own error codes. If any of those are encountered, no data is returned, but the error code is printed to the user. These errors do not stop the program from attempting future updates.

2.1.5 persistence.py

This file contains the necessary functions to delete, create, update, and print database tables. Delete and print functions require only the table name to be specified. Create requires the source URL and the table name. Update requires the new data provided from the fetcher.py file and the table name. These functions are called in other files to manage the SQL database as desired.

2.1.6 scheduler.py

This file contains two functions to automatically update the SQL table on a predetermined schedule, both of which require the source URL and table name as inputs. These functions check the error status of the update and only modify the table if data was successfully read from the source.