PowerwallToChords

Fetch powerwall data from Tesla using the unofficial Tesla Owner API, average it, and send to CHORDS.

PowerwallToChords is built on the wonderful pypowerwall package. See those docs for information on using the package.

Pypowerwall has a python module for accessing the Tesla API, and a containerized web service with a Grafana dashboard. We are only using the python module.

As the docs explain, there are different ways of accessing the powerwall data:

1. *Direct:* Direct access to your powerwall on your local network. This is not available for Powerwall 3s, and even then would require static routing to the fixed Powerwall network IP (192.168.91).
2. *Fleet:* Using the Tesla Fleet API. This apparently is for installers and reseller servicers, and requires you to create an account with Tesla. Supposedly it can be done, but requires some fiddling. The upside is that huge number of additional metrics are available via the *fleet* API.
3. *Cloud:* Using the Tesla Owners API. This requires an authetication key.

We are using the *Cloud* method. The docs walk you through the process of obtaining the authentication information. It gets stored in two files:
`.pypowerwall.auth`
`.pypowerwall.site`
Of course, do not make these files publically accessible.

Running

The Python *zoneinfo* module is required, which wasn't available until Python 3.9. You may need to run in a venv.

```
python3 powerwallToChords.py  -h
usage: powerwallToChords.py [-h] -c CONFIG [--debug]

options:
  -h, --help            show this help message and exit
  -c CONFIG, --config CONFIG
                        Path to json configuration file to use.
  --debug               Enable debug logging
```

Configuration

The configuration file for *PowerwallToChords* looks like:

```
{
    "powerwalltochords": {
        "owner_email": "owner@gmail.com",
        "timezone": "America/Denver",
        "pw_auth_path": "~/",
        "poll_secs": 6,
        "avg_count": 11,
        "debug": false
    },
    "chords": {
        "chords_host": "wx.myhost.com",
        "api_email": "myemail@gmail.com",
        "api_key": "XXXXXXXXXXXXXXXXX",
```

```
        "instrument_id": "12345"
    }
}
```

- `owner_email`: The email for your Tesla account.
- `pw_auth_path`: Directory where the Tesla authenication files are located.
- `poll_secs`: Tesla is queried every `poll_secs`. (*pypowerwall* docs say that data is cached so that calls will not be made any more often than 5s).
- `avg_count`: Number of values that are collected, averaged, and then sent to CHORDS.

Linux Service

linux/powerwalltochords.service provides a typical service definition. Note how it is using a venv to run a suitable version of python.