main 5 Branches 0 Tags Go to file Go to file Code

comes Merge pull request #8 from notnullxyz/main 2310145 · 10 months ago 29 Commits

.github/workflows	Update pint.yml	10 months ago
app	PHP Linting (Pint)	10 months ago
bootstrap	porting to laravel-zero	4 years ago
config	upgrade to 10.x	last year
tests	upgrade to 10.x	last year
.editorconfig	upgrade to 10.x	last year
.gitattributes	porting to laravel-zero	4 years ago
.gitignore	porting to laravel-zero	4 years ago
LICENSE	Initial commit	4 years ago
README.md	porting to laravel-zero	4 years ago
box.json	restore box.json	last year
composer.json	Fix wind index. Add debug logging. Ge…	10 months ago
composer.lock	Fix wind index. Add debug logging. Ge…	10 months ago
ecowitt2weewx	porting to laravel-zero	4 years ago
phpunit.xml.dist	upgrade to 10.x	last year

README License

Restore weewx Database from Ecowitt.net

PRs welcome!

General

In case your weather station send all data to ecowitt.net and you lost your weewx database, this huge cannonball will help you to reintegrate your data. build with https://laravel-zero.com

The steps are easy.

1. Export and Convert

This simple tool will login into your ecowitt.net account, fetch all available devices and download all available data for the range between `startdate` and `enddate`.

```
./ecowitt2weewx export --user <ecowitt username> --pass <ecowitt password> start_date end_date
```

example

```
./ecowitt2weewx export --user foo@example.org --pass 12345 2020-01-01 2020-12-31
```

Details see source file: `/app/Commands/EcowittExportCommand.php`

This script will generate a CSV File which is compatible with wee_import config in step 2. The file will be generated in `ecowitt_<device_id>.csv`.

2. import

Import your data with wee_import. Sample config below

Sample Import File: weewx_import.conf:

```
# EXAMPLE CONFIGURATION FILE FOR IMPORTING FROM CSV FILES
#
# Copyright (c) 2009-2019 Tom Keffer <tkeffer@gmail.com> and Gary Roderick.
# See the file LICENSE.txt for your rights.

###############################################################################
```

About

convert data from ecowitt to weewx

#weewx #ecowitt

Readme
Unlicense license
Activity
5 stars
3 watching
4 forks
Report repository

Releases

No releases published

Packages

No packages published

Contributors 3

comes Jeremias Wolff
notnullxyz Marlon van der Linde
Djblaik David Blaik

Languages

PHP 100.0%

```
# Specify the source. Available options are:
#   CSV - import obs from a single CSV format file
#   WU - import obs from a Weather Underground PWS history
#   Cumulus - import obs from a one or more Cumulus monthly log files
#   WD - import obs from a one or more WD monthly log files
# Format is:
#   source = (CSV | WU | Cumulus)
source = CSV

##########################################################################################

[CSV]
    # Parameters used when importing from a CSV file

    # Path and name of our CSV source file. Format is:
    #   file = full path and filename
    file = /home/pi/input.csv

    # The character used to separate fields. Format is:
    #   delimiter = <single character>
    # Default is , (comma).
    delimiter = ","

    # If there is no mapped interval field how will the interval field be
    # determined for the imported records. Available options are:
    #   derive - Derive the interval field from the timestamp of successive
    #            records. This setting is best used when the imported records
    #            are equally spaced in time and there are no missing records.
    #   conf   - Use the interval setting from weewx.conf. This setting is
    #            best used if the records to be imported have been produced by
    #            WeeWX using the same archive interval as set in weewx.conf on
    #            this machine.
    #   x      - Use a fixed interval of x minutes for every record. This
    #            setting is best used if the records to be imported are
    #            equally based in time but there are some missing records.
    #
    # Note: If there is a mapped interval field then this setting will be
    #       ignored.
    # Format is:
    #   interval = (derive | conf | x)
    interval = derive

    # Should the [StdQC] max/min limits in weewx.conf be applied to the
    # imported data. This may be useful if the source has extreme values that
    # are clearly incorrect for some observations. Available options are:
    #   True  - weewx.conf [StdQC] max/min limits are applied.
    #   False - weewx.conf [StdQC] max/min limits are not applied.
    # Format is:
    #   qc = (True | False)
    qc = True

    # Should any missing derived observations be calculated from the imported
    # data if possible. Available options are:
    #   True  - Any missing derived observations are calculated.
    #   False - Any missing derived observations are not calculated.
    # Format is:
    #   calc_missing = (True | False)
    calc_missing = True

    # Specify how imported data fields that contain invalid data (eg a numeric
    # field containing non-numeric data) are handled. Available options are:
    #   True  - The invalid data is ignored, the WeeWX target field is set to
    #           None and the import continues.
    #   False - The import is halted.
    # Format is:
    #   ignore_invalid_data = (True | False)
    # Default is True.
    ignore_invalid_data = True

    # Imported records are written to archive in transactions of tranche
    # records at a time. Increase for faster throughput, decrease to reduce
    # memory requirements. Format is:
    #   tranche = x
    # where x is an integer
    tranche = 250

    # Specify whether a UV sensor was used to produce any UV observations.
    # Available options are:
    #   True  - UV sensor was used and UV data will be imported.
    #   False - UV sensor was not used and any UV data will not be imported.
    #           UV fields will be set to None/NULL.
    # For a CSV import UV_sensor should be set to False if a UV sensor was
    # NOT present when the import data was created. Otherwise it may be set to
    # True or omitted. Format is:
    #   UV_sensor = (True | False)
    UV_sensor = True

    # Specify whether a solar radiation sensor was used to produce any solar
    # radiation observations. Available options are:
    #   True  - Solar radiation sensor was used and solar radiation data will
    #           be imported.
    #   False - Solar radiation sensor was not used and any solar radiation
    #           data will not be imported. radiation fields will be set to
    #           None/NULL.
    # For a CSV import solar_sensor should be set to False if a solar radiation
    # sensor was NOT present when the import data was created. Otherwise it may
```

```
    # be set to True or omitted. Format is:
    #   solar_sensor = (True | False)
    solar_sensor = True

    # Date-time format of CSV field from which the WeeWX archive record
    # dateTime field is to be extracted. wee_import first attempts to interpret
    # date/time info in this format, if this fails it then attempts to
    # interpret it as a timestamp and if this fails it then raises an error.
    # Uses Python strptime() format codes.
    # raw_datetime_format = Python strptime() format string
    raw_datetime_format = %Y-%m-%d %H:%M

    # Does the imported rain field represent the total rainfall since the last
    # record or a cumulative value. Available options are:
    #   discrete   - rain field represents total rainfall since last record
    #   cumulative - rain field represents a cumulative rainfall reset at
    #                midnight
    # rain = (discrete | cumulative)
    rain = cumulative

    # Lower and upper bounds for imported wind direction. It is possible,
    # particularly for a calculated direction, to have a value (eg -45) outside
    # of the WeeWX limits (0 to 360 inclusive). Format is:
    #
    # wind_direction = lower,upper
    #
    # where :
    #   lower is the lower limit of acceptable wind direction in degrees
    #   (may be negative)
    #   upper is the upper limit of acceptable wind direction in degrees
    #
    # Imported values from lower to upper will be normalised to the range 0 to
    # 360. Values outside of the parameter range will be stored as None.
    # Default is -360,360.
    wind_direction = -360,360

    # Map CSV record fields to WeeWX archive fields. Format is:
    #
    #   weewx_archive_field_name = csv_field_name, weewx_unit_name
    #
    # where:
    #   weewx_archive_field_name - An observation name in the WeeWX database
    #                              schema.
    #   csv_field_name           - The name of a field from the CSV file.
    #   weewx_unit_name          - The name of the units, as defined in WeeWX,
    #                              used by csv_field_name. wee_import will do
    #                              the necessary conversions to the unit system
    #                              used by the WeeWX archive.
    # For example,
    #   outTemp = Temp, degree_C
    # would map the CSV field Temp, in degrees C, to the archive field outTemp.
    #
    # A mapping for WeeWX field dateTime is mandatory and the WeeWX unit name
    # for the dateTime mapping must be unix_epoch. For example,
    #   dateTime = csv_date_and_time, unix_epoch
    # would map the CSV field csv_date_and_time to the WeeWX dateTime field with
    # the csv_date_and_time field being interpreted first using the format
    # specified at the raw_datetime_format config option and if that fails as a
    # unix epoch timestamp.
    #
    # Field mapping to the WeeWX usUnits archive field is currently not
    # supported. If a usUnits field exists in the CSV data it should not be
    # mapped, rather WeeWX unit names should included against each field to be
    # imported in the field map.
    #
    # WeeWX archive fields that do not exist in the CSV data may be omitted.
    # Any omitted fields that are derived (eg dewpoint) may be calculated
    # during import using the equivalent of the WeeWX StdWXCalculate service
    # through setting the calc-missing parameter above.
    [[FieldMap]]
        dateTime    = date_and_time
        interval    =
        barometer   = pressure_rel, hPa
        pressure    = pressure_abs, hPa
        altimeter   =
        inTemp      = temp_in, degree_C
        outTemp     = temp_out, degree_C
        inHumidity  = humid_in, percent
        outHumidity = humid_out, percent
        windSpeed   = wind, meter_per_second
        windDir     = wind_dir, degree_compass
        windGust    = wind_gust, meter_per_second
        windGustDir =
        rainRate    = rain, mm_per_hour
        rain        = rain_daily, mm
        dewpoint    = temp_out_dew, degree_C
        windchill   = temp_out_gust, degree_C
        heatindex   =
        ET          =
        radiation   =
        UV          =
```

DONE!

regenerate / restart weewx daemon and check your data!