weatherlink-live-local-python

Release 0.2.0

Lukas Berbuer

Dec 23, 2020

LIBRARY DOCUMENTATION

Python library to read weather data from a Davis WeatherLink Live station + connected sensors (e.g. Vantage Vue) using the local API.

API specification: https://weatherlink.github.io/weatherlink-live-local-api/

weatherlink_live_local	WeatherLink Live Local Python API.

CHAPTER

ONE

WEATHERLINK_LIVE_LOCAL

WeatherLink Live Local Python API.

Functions

discover([timeout])	Discover all WeatherLink Live services on local network(s).
get_conditions(ip[, port, timeout])	Read conditions from WeatherLink Live device + all connected sensors.
set_units([temperature, pressure, rain, ...])	Set desired units for *get_conditions* command.

1.1 discover

weatherlink_live_local.**discover**(*timeout=1*)

Discover all WeatherLink Live services on local network(s).

Parameters **timeout** (int) – Timeout in seconds

Return type List[*ServiceInfo*]

Returns List of found services

1.2 get_conditions

weatherlink_live_local.**get_conditions**(*ip*, *port=80*, *timeout=1*)

Read conditions from WeatherLink Live device + all connected sensors.

Parameters

- **ip** (str) – IP address of WeatherLink Live device. Use *discover* function to find devices with their IP address in the local network.
- **port** (int) – Port number of HTTP interface, should be 80
- **timeout** (int) – Maximum time to listen for WeatherLink Live services

Return type *Conditions*

Returns Conditions of all available sensors

1.3 set_units

weatherlink_live_local.**set_units**(*temperature=None*, *pressure=None*, *rain=None*, *wind_speed=None*)

Set desired units for *get_conditions* command.

Parameters

- **temperature** (Optional[*TemperatureUnit*]) – Temperature unit
- **pressure** (Optional[*PressureUnit*]) – Pressure unit
- **rain** (Optional[*RainUnit*]) – Rain amount unit
- **wind_speed** (Optional[*WindSpeedUnit*]) – Wind speed unit

Example

```
>>> import weatherlink_live_local as wlll
>>> # change only a single unit
>>> wlll.set_units(temperature=wlll.units.TemperatureUnit.CELSIUS)
>>> # change multiple units at once
>>> wlll.set_units(
>>>     pressure=wlll.units.PressureUnit.HECTOPASCAL,
>>>     rain=wlll.units.RainUnit.MILLIMETER,
>>> )
```

Modules

weatherlink_live_local.conditions	Datatypes to gather device-specific sensor data / conditions.
weatherlink_live_local.discovery	Zeroconf-based discovery of WeatherLink Live services/devices in the local network(s).
weatherlink_live_local.units	Units and conversion from default imperial system.

1.4 conditions

Datatypes to gather device-specific sensor data / conditions.

Classes

BarometricConditions(lsid, bar_sea_level, ...)	Barometric conditions of WeatherLink Live station.
Conditions(timestamp, inside, barometric, ...)	Gathered conditions of all available sensors.
InsideConditions(lsid, temp, hum, dew_point, ...)	Inside conditions of WeatherLink Live station.
MoistureTemperatureConditions(txid, ...)	Conditions of leaf & soil moisture/temperature station.
RadioReceptionState(value)	Transmitter radio reception state.
SensorSuiteConditions(txid, rx_state, ...)	Conditions of integrated sensor suite (ISS), e.g.

1.4.1 BarometricConditions

class weatherlink_live_local.conditions.**BarometricConditions**(*lsid*, *bar_sea_level*, *bar_trend*, *bar_absolute*)

Barometric conditions of WeatherLink Live station.

Data structure for *data_structure_type = 3*

__init__(*lsid*, *bar_sea_level*, *bar_trend*, *bar_absolute*)
Initialize self. See help(type(self)) for accurate signature.

Methods

__init__(lsid, bar_sea_level, bar_trend, ...)	Initialize self.
from_dict(json_data)	

bar_sea_level: Optional[float]
Most recent bar sensor reading with elevation adjustment [*PressureUnit*]

bar_trend: Optional[float]
Current 3 hour bar trend [*PressureUnit*]

bar_absolute: Optional[float]
Raw bar sensor reading [*PressureUnit*]

1.4.2 Conditions

class weatherlink_live_local.conditions.**Conditions**(*timestamp*, *inside*, *barometric*, *moisture_temperature_stations*, *integrated_sensor_suites*)

Gathered conditions of all available sensors.

Returned by *get_conditions* function.

__init__(*timestamp*, *inside*, *barometric*, *moisture_temperature_stations*, *integrated_sensor_suites*)
Initialize self. See help(type(self)) for accurate signature.

Methods

__init__(timestamp, inside, barometric, ...)	Initialize self.

timestamp: datetime.datetime
Timestamp

inside: *weatherlink_live_local.conditions.InsideConditions*
Inside conditions of WeatherLink Live station

barometric: *weatherlink_live_local.conditions.BarometricConditions*
Barometric conditions of WeatherLink Live station

moisture_temperature_stations: List[*weatherlink_live_local.conditions.MoistureTempera*
Conditions of leaf & soil moisture/temperature station(s)

integrated_sensor_suites: List[*weatherlink_live_local.conditions.SensorSuiteCondition*

Conditions of integrated sensor suite(s), e.g. Vantage Vue

1.4.3 InsideConditions

class weatherlink_live_local.conditions.**InsideConditions**(*lsid*, *temp*, *hum*, *dew_point*, *heat_index*)

Inside conditions of WeatherLink Live station.

Data structure for *data_structure_type = 4*

__init__(*lsid*, *temp*, *hum*, *dew_point*, *heat_index*)
Initialize self. See help(type(self)) for accurate signature.

Methods

__init__(lsid, temp, hum, dew_point, heat_index)	Initialize self.
from_dict(json_data)	

temp: Optional[float]
Inside temperature [*TemperatureUnit*]

hum: Optional[float]
Inside humidity [%]

dew_point: Optional[float]
Dew point [*TemperatureUnit*]

heat_index: Optional[float]
Heat index [*TemperatureUnit*]

1.4.4 MoistureTemperatureConditions

class weatherlink_live_local.conditions.**MoistureTemperatureConditions**(*txid*, *rx_state*, *trans_battery_flag*, *lsid*, *temp_1*, *temp_2*, *temp_3*, *temp_4*, *moist_soil_1*, *moist_soil_2*, *moist_soil_3*, *moist_soil_4*, *wet_leaf_1*, *wet_leaf_2*)

Conditions of leaf & soil moisture/temperature station.

Data structure for *data_structure_type = 2*

__init__(*txid*, *rx_state*, *trans_battery_flag*, *lsid*, *temp_1*, *temp_2*, *temp_3*, *temp_4*, *moist_soil_1*, *moist_soil_2*, *moist_soil_3*, *moist_soil_4*, *wet_leaf_1*, *wet_leaf_2*)
Initialize self. See help(type(self)) for accurate signature.

Methods

__init__(txid, rx_state, trans_battery_flag, ...)	Initialize self.
from_dict(json_data)	

temp_1: Optional[float]
Temperature slot 1 [*TemperatureUnit*]

temp_2: Optional[float]
Temperature slot 2 [*TemperatureUnit*]

temp_3: Optional[float]
Temperature slot 3 [*TemperatureUnit*]

temp_4: Optional[float]
Temperature slot 4 [*TemperatureUnit*]

moist_soil_1: Optional[float]
Moisture soil slot 1 [cb]

moist_soil_2: Optional[float]
Moisture soil slot 2 [cb]

moist_soil_3: Optional[float]
Moisture soil slot 3 [cb]

moist_soil_4: Optional[float]
Moisture soil slot 4 [cb]

wet_leaf_1: Optional[float]
Leaf wetness slot 1

wet_leaf_2: Optional[float]
Leaf wetness slot 2

1.4.5 RadioReceptionState

class weatherlink_live_local.conditions.**RadioReceptionState**(*value*)
Transmitter radio reception state.

__init__()
Initialize self. See help(type(self)) for accurate signature.

Attributes

SCANNING	Transmitter has not been acquired yet, or we've lost it (more than 15 missed packets in a row)
SYNCED	Transmitter has been acquired, but we have missed 1-14 packets in a row
SYNCED_TRACKING	Transmitter has been acquired and is actively being received

SYNCED_TRACKING = 0
Transmitter has been acquired and is actively being received

SYNCED = 1

Transmitter has been acquired, but we have missed 1-14 packets in a row

SCANNING = 2
Transmitter has not been acquired yet, or we've lost it (more than 15 missed packets in a row)

1.4.6 SensorSuiteConditions

class weatherlink_live_local.conditions.**SensorSuiteConditions**(*txid*, *rx_state*, *trans_battery_flag*, *lsid*, *temp*, *hum*, *dew_point*, *wet_bulb*, *heat_index*, *wind_chill*, *thw_index*, *thsw_index*, *wind_speed_last*, *wind_speed_avg_last_1_min*, *wind_speed_avg_last_2_min*, *wind_speed_avg_last_10_min*, *wind_speed_hi_last_2_min*, *wind_speed_hi_last_10_min*, *wind_dir_last*, *wind_dir_scalar_avg_last_1_min*, *wind_dir_scalar_avg_last_2_min*, *wind_dir_scalar_avg_last_10_min*, *wind_dir_at_hi_speed_last_2_min*, *wind_dir_at_hi_speed_last_10_min*, *rainfall_last_60_min*, *rainfall_last_24_hr*, *rainfall_daily*, *rainfall_monthly*, *rainfall_year*, *rain_rate_last*, *rain_rate_hi_last_1_min*, *rain_rate_hi_last_15_min*, *rain_storm_last*, *rain_storm_last_start_at*, *rain_storm_last_end_at*, *solar_rad*, *uv_index*)

Conditions of integrated sensor suite (ISS), e.g. Vantage Vue.

Data structure for *data_structure_type = 1*

__init__(*txid*, *rx_state*, *trans_battery_flag*, *lsid*, *temp*, *hum*, *dew_point*, *wet_bulb*, *heat_index*, *wind_chill*, *thw_index*, *thsw_index*, *wind_speed_last*, *wind_speed_avg_last_1_min*, *wind_speed_avg_last_2_min*, *wind_speed_avg_last_10_min*, *wind_speed_hi_last_2_min*, *wind_speed_hi_last_10_min*, *wind_dir_last*, *wind_dir_scalar_avg_last_1_min*, *wind_dir_scalar_avg_last_2_min*, *wind_dir_scalar_avg_last_10_min*, *wind_dir_at_hi_speed_last_2_min*, *wind_dir_at_hi_speed_last_10_min*, *rainfall_last_60_min*, *rainfall_last_24_hr*, *rainfall_daily*, *rainfall_monthly*, *rainfall_year*, *rain_rate_last*, *rain_rate_hi_last_1_min*, *rain_rate_hi_last_15_min*, *rain_storm_last*, *rain_storm_last_start_at*, *rain_storm_last_end_at*, *solar_rad*, *uv_index*)
Initialize self. See help(type(self)) for accurate signature.

Methods

__init__(txid, rx_state, trans_battery_flag, ...)	Initialize self.
from_dict(json_data)	

temp: Optional[float]
Temperature [*TemperatureUnit*]

hum: Optional[float]
Humidity [%]

dew_point: Optional[float]
Dew point [*TemperatureUnit*]

wet_bulb: Optional[float]
Wet bulb [*TemperatureUnit*]

heat_index: Optional[float]
Heat index [*TemperatureUnit*]

wind_chill: Optional[float]
Wind chill [*TemperatureUnit*]

thw_index: Optional[float]
THW index [*TemperatureUnit*]

thsw_index: Optional[float]
THSW index [*TemperatureUnit*]

wind_speed_last: Optional[float]
Most recent wind speed [*WindSpeedUnit*]

wind_speed_avg_last_1_min: Optional[float]
Average wind speed over last 1 min [*WindSpeedUnit*]

wind_speed_avg_last_2_min: Optional[float]
Average wind speed over last 2 min [*WindSpeedUnit*]

wind_speed_avg_last_10_min: Optional[float]
Average wind speed over last 10 min [*WindSpeedUnit*]

wind_speed_hi_last_2_min: Optional[float]
Maximum wind speed over last 2 min [*WindSpeedUnit*]

wind_speed_hi_last_10_min: Optional[float]
Maximum wind speed over last 10 min [*WindSpeedUnit*]

wind_dir_last: Optional[float]
Wind direction [°]

wind_dir_scalar_avg_last_1_min: Optional[float]
Average wind direction over last 1 min [°]

wind_dir_scalar_avg_last_2_min: Optional[float]
Average wind direction over last 2 min [°]

wind_dir_scalar_avg_last_10_min: Optional[float]
Average wind direction over last 10 min [°]

wind_dir_at_hi_speed_last_2_min: Optional[float]
Gust wind direction over last 2 min [°]

wind_dir_at_hi_speed_last_10_min: Optional[float]
Gust wind direction over last 10 min [°]

rainfall_last_60_min: Optional[float]
Total rain for last 60 min [*RainUnit*]

rainfall_last_24_hr: Optional[float]
Total rain for last 24 hours [*RainUnit*]

rainfall_daily: Optional[float]
Total rain since local midnight [*RainUnit*]

rainfall_monthly: Optional[float]
Total rain since first of month [*RainUnit*]

rainfall_year: Optional[float]
Total rain since first of user-chosen month at local midnight [*RainUnit*]

rain_rate_last: Optional[float]
Rain rate [*RainUnit*/hour]

rain_rate_hi_last_1_min: Optional[float]
Highest rain rate over last 1 min [*RainUnit*/hour]

rain_rate_hi_last_15_min: Optional[float]
Highest rain rate over last 15 min [*RainUnit*/hour]

rain_storm_last: Optional[float]
Total rain since last 24 hour long break in rain [*RainUnit*]

rain_storm_last_start_at: Optional[datetime.datetime]
Timestamp of last rain storm start

rain_storm_last_end_at: Optional[datetime.datetime]
Timestamp of last rain storm start

solar_rad: Optional[float]
Solar radiation [W/m^2]

uv_index: Optional[float]
UV index

1.5 discovery

Zeroconf-based discovery of WeatherLink Live services/devices in the local network(s).

Classes

Discovery()	Discovery for all WeatherLink Live services on local network(s).
ServiceInfo(name, ip_addresses, port)	WeatherLink Live service information.

1.5.1 Discovery

class weatherlink_live_local.discovery.**Discovery**

Discovery for all WeatherLink Live services on local network(s).

__init__()

Initialize self. See help(type(self)) for accurate signature.

Methods

__init__()	Initialize self.
add_service(zc, type_, name)	**rtype** None
find(timeout)	**rtype** List[*ServiceInfo*]
remove_service(zc, type_, name)	**rtype** None
update_service(zc, type_, name)	**rtype** None

Attributes

TYPE

1.5.2 ServiceInfo

class weatherlink_live_local.discovery.**ServiceInfo**(*name: str*, *ip_addresses: List[str]*, *port: int*)

WeatherLink Live service information.

__init__()
Initialize self. See help(type(self)) for accurate signature.

Methods

__init__()	Initialize self.
count(value, /)	Return number of occurrences of value.
index(value[, start, stop])	Return first index of value.

Attributes

ip_addresses	IP address of device, usually only one
name	Unique name of service
port	Port number, usually 80

property name
Unique name of service

property ip_addresses
IP address of device, usually only one

property port
Port number, usually 80

count(*value*, */*)
Return number of occurrences of value.

index(*value*, *start=0*, *stop=9223372036854775807*, */*)
Return first index of value.

Raises ValueError if the value is not present.

1.6 units

Units and conversion from default imperial system.

Functions

convert_pressure(inhg)	Convert imperial pressure (inches of mercury) to selected unit.
convert_rain(inch)	Convert imperial rain amount (inch) to selected unit.
convert_temperature(fahrenheit)	Convert imperial temperature (Fahrenheit) to selected unit.
convert_wind_speed(mph)	Convert imperial wind speed (miles per hour) to selected unit.

1.6.1 convert_pressure

weatherlink_live_local.units.**convert_pressure**(*inhg*)

Convert imperial pressure (inches of mercury) to selected unit.

Return type Optional[float]

1.6.2 convert_rain

weatherlink_live_local.units.**convert_rain**(*inch*)

Convert imperial rain amount (inch) to selected unit.

Return type Optional[float]

1.6.3 convert_temperature

weatherlink_live_local.units.**convert_temperature**(*fahrenheit*)

Convert imperial temperature (Fahrenheit) to selected unit.

Return type Optional[float]

1.6.4 convert_wind_speed

weatherlink_live_local.units.**convert_wind_speed**(*mph*)

Convert imperial wind speed (miles per hour) to selected unit.

Return type Optional[float]

Classes

PressureUnit(value)	Pressure unit for barometric sensor values.
RainUnit(value)	Rain quantity unit.
TemperatureUnit(value)	Temperature unit.
WindSpeedUnit(value)	Wind speed unit.

1.6.5 PressureUnit

class weatherlink_live_local.units.**PressureUnit**(*value*)

Pressure unit for barometric sensor values.

__init__()

Initialize self. See help(type(self)) for accurate signature.

Attributes

HECTOPASCAL	Hecto-pascal hPa
INCH_MERCURY	Inches of mercury in Hg

HECTOPASCAL = 1

Hecto-pascal hPa

INCH_MERCURY = 2

Inches of mercury in Hg

1.6.6 RainUnit

class weatherlink_live_local.units.**RainUnit**(*value*)

Rain quantity unit.

__init__()

Initialize self. See help(type(self)) for accurate signature.

Attributes

INCH	Inch
MILLIMETER	Millimeter mm

MILLIMETER = 1

Millimeter mm

INCH = 2

Inch

1.6.7 TemperatureUnit

class weatherlink_live_local.units.**TemperatureUnit**(*value*)

Temperature unit.

__init__()

Initialize self. See help(type(self)) for accurate signature.

Attributes

CELSIUS	Degree Celsius °C
FAHRENHEIT	Degree Fahrenheit °F

CELSIUS = 1

Degree Celsius °C

FAHRENHEIT = 2

Degree Fahrenheit °F

1.6.8 WindSpeedUnit

class weatherlink_live_local.units.**WindSpeedUnit**(*value*)

Wind speed unit.

__init__()

Initialize self. See help(type(self)) for accurate signature.

Attributes

METER_PER_SECOND	Meter per seconds m/s
MILES_PER_HOUR	Miles per hour mi/h

METER_PER_SECOND = 1

Meter per seconds m/s

MILES_PER_HOUR = 2

Miles per hour mi/h

CHAPTER

TWO

CHANGELOG

All notable changes to this project will be documented in this file.

The format is based on Keep a Changelog, and this project adheres to Semantic Versioning.

2.1 Unreleased

2.2 0.2.0 - 2020-12-23

2.2.1 Added

- CI with GitHub Actions

2.2.2 Changed

- Remove imports of `conditions` and `discovery` module in global namespace

2.2.3 Fixed

- Project links in setup.py

2.3 0.1.0 - 2020-12-23

Initial public release

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex

PYTHON MODULE INDEX

W

INDEX

Symbols

B

C

D

F

G

H

I

M

N

P

R

S

T

U

W