libtado Documentation

Max Rosin

Oct 04, 2022

Getting-started

libtado is a simple Python library that provides methods to control the smart heating devices from the German company tado GmbH[1]. It uses the undocumented REST API of their website.

The library is in NO way connected to tado GmbH and is not officially supported by them!

The source code is hosted on GitHub[2]. Feel free to report issues or open pull requests.

[1] https://www.tado.com
[2] https://github.com/germainlefebvre4/libtado

CHAPTER 1

Usage

After you installed libtado you can easily test it by using the included *command line client* like this:

```
tado --username USERNAME --password PASSWORD whoami
```

To use the library in your own code you can start with this:

```
import tado.api
t = tado.api('Username', 'Password')
print(t.get_me())
```

Check out *all available API methods* to learn what you can to with libtado.

CHAPTER 2

Installation

You can download the official library on Pypi using *pip*.

```
pip install libtado.
```

It cannot be easier ;)

CHAPTER 3

Frequently Asked Questions

3.1 How to retrieve the client secret

Option #1: Do nothing

The library wil automatically retrieve the client secret on following the steps at *Options #2*.

Option #2: From the application 'env.js'

Retrieve the *CLIENT_SECRET* before running the script otherwise you will get a *401 Unauthorized Access*. The latest *CLIENT_SECRET* can be found at [https://my.tado.com/webapp/env.js{]}(https://my.tado.com/webapp/env.js). It will look something like this:

Option #3: From the developer mode

An alternative way to get your *CLIENT_SECRET* is to enable the Developper Mode when logging in and catch the Headers. You will find the form data like this :

```
client_id: tado-web-app
client_secret: fndskjnjzkefjNFRNkfKJRNFKRENkjnrek
grant_type: password
password: MyBeautifulPassword
scope: home.user
username: email@example.com
```

Then you just have to get the value in the attribute *client_secret*. You will need it to connect to your account through Tado APIs. The *client_secret* never dies so you can base your script on it.

CHAPTER 4

Command Line Client

```
Usage: tado [OPTIONS] COMMAND [ARGS]...

  This script provides a command line client for the Tado API.

  You can use the environment variables TADO_USERNAME and TADO_PASSWORD
  instead of the command line options.

  Call 'tado COMMAND --help' to see available options for subcommands.

Options:
  -u, --username TEXT       Tado username  [required]
  -p, --password TEXT       Tado password  [required]
  -c, --client-secret TEXT  Tado password  [optional]
  -h, --help                Show this message and exit.

Commands:
  capabilities        Display the capabilities of a zone.
  devices             Display all devices.
  early_start         Display or change the early start feature of a zone.
  end_manual_control  End manual control of a zone.
  home                Display information about your home.
  mobile              Display all mobile devices.
  set_temperature     Set the desired temperature of a zone.
  users               Display all users of your home.
  whoami              Tell me who the Tado API thinks I am.
  zone                Get the current state of a zone.
  zones               Get configuration information about all zones.
```

CHAPTER 5

API Class Methods

Most of the functions receive JSON from the API and translate it to a Python dictionary or a list of Python dictionaries. So the return value will depend on the behaviour of the API.

5.1 Getter

5.1.1 Air comfort

class libtado.api.Tado.**get_air_comfort**

Get all zones of your home.

Returns A list of dictionaries with all your zones.

Return type list

Example

```
{
    "freshness":{
        "value":"FAIR",
        "lastOpenWindow":"2020-09-04T10:38:57Z"
    },
    "comfort":[
        {
            "roomId":1,
            "temperatureLevel":"COMFY",
            "humidityLevel":"COMFY",
            "coordinate":{
                "radial":0.36,
                "angular":323
            }
        },
```

(continues on next page)

(continued from previous page)


```
        {
            "roomId":4,
            "temperatureLevel":"COMFY",
            "humidityLevel":"COMFY",
            "coordinate":{
                "radial":0.43,
                "angular":324
            }
        }
    ]
}
```


5.1.2 Air comfort geoloc

class libtado.api.Tado.**get_air_comfort_geoloc**

Get all zones of your home.

Parameters

- **latitude** (*float*) – The latitude of the home.
- **longitude** (*float*) – The longitude of the home.

Returns A dict of lists of dictionaries with all your rooms.

Return type list

Example


```
{
    "roomMessages":[
        {
            "roomId":4,
            "message":"Bravo ! L'air de cette pièce est proche de la perfection.",
            "visual":"success",
            "link":null
        },
        {
            "roomId":1,
            "message":"Continuez à faire ce que vous faites ! L'air de cette␣
↪pièce est parfait.",
            "visual":"success",
            "link":null
        }
    ],
    "outdoorQuality":{
        "aqi":{
            "value":81,
            "level":"EXCELLENT"
        },
        "pollens":{
            "dominant":{
                "level":"LOW"
            },
            "types":[
```


(continues on next page)

(continued from previous page)


```
                {
                    "localizedName":"Graminées",
                    "type":"GRASS",
                    "localizedDescription":"Poaceae",
                    "forecast":[
                        {
                            "localizedDay":"Auj.",
                            "date":"2020-09-06",
                            "level":"NONE"
                        },
                        {
                            "localizedDay":"Lun",
                            "date":"2020-09-07",
                            "level":"NONE"
                        },
                        {
                            "localizedDay":"Mar",
                            "date":"2020-09-08",
                            "level":"NONE"
                        }
                    ]
                },
                {
                    "localizedName":"Herbacées",
                    "type":"WEED",
                    "localizedDescription":"Armoise, Ambroisie, Pariétaire",
                    "forecast":[
                        {
                            "localizedDay":"Auj.",
                            "date":"2020-09-06",
                            "level":"NONE"
                        },
                        {
                            "localizedDay":"Lun",
                            "date":"2020-09-07",
                            "level":"NONE"
                        },
                        {
                            "localizedDay":"Mar",
                            "date":"2020-09-08",
                            "level":"NONE"
                        }
                    ]
                },
                {
                    "localizedName":"Arbres",
                    "type":"TREE",
                    "localizedDescription":"Aulne, Frêne, Bouleau, Noisetier,␣
↪Cyprès, Olivier",
                    "forecast":[
                        {
                            "localizedDay":"Auj.",
                            "date":"2020-09-06",
                            "level":"NONE"
                        },
                        {
                            "localizedDay":"Lun",
```


(continues on next page)

(continued from previous page)


```
                        "date":"2020-09-07",
                        "level":"NONE"
                    },
                    {
                        "localizedDay":"Mar",
                        "date":"2020-09-08",
                        "level":"NONE"
                    }
                ]
            }
        ]
    },
    "pollutants":[
        {
            "localizedName":"Matière particulaire",
            "scientificName":"PM<sub>10</sub>",
            "level":"EXCELLENT",
            "concentration":{
                "value":8.75,
                "units":"µg/m<sup>3</sup>"
            }
        },
        {
            "localizedName":"Matière particulaire",
            "scientificName":"PM<sub>2.5</sub>",
            "level":"EXCELLENT",
            "concentration":{
                "value":5.04,
                "units":"µg/m<sup>3</sup>"
            }
        },
        {
            "localizedName":"Ozone",
            "scientificName":"O<sub>3</sub>",
            "level":"EXCELLENT",
            "concentration":{
                "value":23.86,
                "units":"ppb"
            }
        },
        {
            "localizedName":"Dioxyde de soufre",
            "scientificName":"SO<sub>2</sub>",
            "level":"EXCELLENT",
            "concentration":{
                "value":1.19,
                "units":"ppb"
            }
        },
        {
            "localizedName":"Monoxyde de carbone",
            "scientificName":"CO",
            "level":"EXCELLENT",
            "concentration":{
                "value":266.8,
                "units":"ppb"
            }
```


(continues on next page)

(continued from previous page)

```
            },
            {
                "localizedName":"Dioxyde d'azote",
                "scientificName":"NO<sub>2</sub>",
                "level":"EXCELLENT",
                "concentration":{
                    "value":5.76,
                    "units":"ppb"
                }
            }
        ]
    }
}
```

5.1.3 Away configuration

class libtado.api.Tado.**get_away_configuration**

Get the away configuration for a zone

Parameters **zone** (*int*) – The zone ID.

Returns dict

5.1.4 Capabilities

class libtado.api.Tado.**get_capabilities**

Parameters **zone** (*int*) – The zone ID.

Returns The capabilities of a tado zone as dictionary.

Return type dict

Example

```
{
  'temperatures': {
    'celsius': {'max': 25, 'min': 5, 'step': 1.0},
    'fahrenheit': {'max': 77, 'min': 41, 'step': 1.0}
  },
  'type': 'HEATING'
}
```

5.1.5 Default overlay

class libtado.api.Tado.**get_default_overlay**

Get the default overlay settings of a zone

Parameters **zone** (*int*) – The zone ID.

Returns dict

Example

```
{
   "terminationCondition": {
     "type": "TADO_MODE"
   }
}
```

5.1.6 Device usage

class libtado.api.Tado.**get_device_usage**

Get all devices of your home with how they are used

Returns All devices of home as list of dictionaries

Return type list

5.1.7 Devices

class libtado.api.Tado.**get_devices**

Returns All devices of the home as a list of dictionaries.

Return type list

Example

```
[
  {
    'characteristics': { 'capabilities': [] },
    'connectionState': {
      'timestamp': '2017-02-20T18:51:47.362Z',
      'value': True
    },
    'currentFwVersion': '25.15',
    'deviceType': 'GW03',
    'gatewayOperation': 'NORMAL',
    'serialNo': 'SOME_SERIAL',
    'shortSerialNo': 'SOME_SERIAL'
  },
  {
    'characteristics': {
      'capabilities': [ 'INSIDE_TEMPERATURE_MEASUREMENT', 'IDENTIFY']
    },
    'connectionState': {
      'timestamp': '2017-01-22T16:03:00.773Z',
      'value': False
    },
    'currentFwVersion': '36.15',
    'deviceType': 'VA01',
    'mountingState': {
      'timestamp': '2017-01-22T15:12:45.360Z',
      'value': 'UNMOUNTED'
    },
```

(continues on next page)

(continued from previous page)


```
    'serialNo': 'SOME_SERIAL',
    'shortSerialNo': 'SOME_SERIAL'
  },
  {
    'characteristics': {
      'capabilities': [ 'INSIDE_TEMPERATURE_MEASUREMENT', 'IDENTIFY']
    },
    'connectionState': {
      'timestamp': '2017-02-20T18:33:49.092Z',
      'value': True
    },
    'currentFwVersion': '36.15',
    'deviceType': 'VA01',
    'mountingState': {
      'timestamp': '2017-02-12T13:34:35.288Z',
      'value': 'CALIBRATED'},
    'serialNo': 'SOME_SERIAL',
    'shortSerialNo': 'SOME_SERIAL'
  },
  {
    'characteristics': {
      'capabilities': [ 'INSIDE_TEMPERATURE_MEASUREMENT', 'IDENTIFY']
    },
    'connectionState': {
      'timestamp': '2017-02-20T18:51:28.779Z',
      'value': True
    },
    'currentFwVersion': '36.15',
    'deviceType': 'VA01',
    'mountingState': {
      'timestamp': '2017-01-12T13:22:11.618Z',
      'value': 'CALIBRATED'
     },
    'serialNo': 'SOME_SERIAL',
    'shortSerialNo': 'SOME_SERIAL'
  }
]
```


5.1.8 Early start

class libtado.api.Tado.**get_early_start**

Get the early start configuration of a zone.

Parameters **zone** (*int*) – The zone ID.

Returns A dictionary with the early start setting of the zone. (True or False)

Return type dict

Example


```
{ 'enabled': True }
```

5.1.9 Energy consumption

class libtado.api.Tado.**get_energy_consumption**
Get enery consumption of your home by range date

Parameters **None.** –

Returns A dict of your energy consumption.

Return type list

Example

{

“tariff”: “0.104 €/kWh”, “unit”: “m3”, “consumptionInputState”: “full”, “customTariff”: false, “currency”: “EUR”, “tariffInfo”:{

“consumptionUnit”: “kWh”, “customTariff”: false, “tariffInCents”: 10.36, “currencySign”: “€”,

“details”:{ “totalCostInCents”: 1762.98, “totalConsumption”: 16.13, “perDay”: [

{ “date”: “2022-09-01”, “consumption”: 0, “costInCents”: 0

},{ “date”: “2022-09-02”, “consumption”: 0, “costInCents”: 0

},{ “date”: “2022-09-03”, “consumption”: 0.04, “costInCents”: 0.4144

}

],

}

}

5.1.10 Energy savings

class libtado.api.Tado.**get_energy_savings**
Get energy savings of your home by month and year

Parameters **None.** –

Returns A dict of your energy savings.

Return type list

Example

{

“coveredInterval”:{ “start”:”2022-08-31T23:48:02.675000Z”, “end”:”2022-09-29T13:10:23.035000Z”

}, “totalSavingsAvailable”:true, “withAutoAssist”:{

“detectedAwayDuration”:{ “value”:56, “unit”:”HOURS”

}, “openWindowDetectionTimes”:9

}, “totalSavingsInThermostaticMode”:{

“value”:0, “unit”:”HOURS”

}, “manualControlSaving”:{

“value”:0, “unit”:”PERCENTAGE”

}, “totalSavings”:{

“value”:6.5, “unit”:”PERCENTAGE”

}, “hideSunshineDuration”:false, “awayDuration”:{

“value”:56, “unit”:”HOURS”

}, “showSavingsInThermostaticMode”:false, “communityNews”:{

“type”:”HOME_COMFORT_STATES”, “states”:[

{ “name”:”humid”, “value”:47.3, “unit”:”PERCENTAGE”

}, {

“name”:”ideal”, “value”:43.1, “unit”:”PERCENTAGE”

}, {

“name”:”cold”, “value”:9.5, “unit”:”PERCENTAGE”

}, {

“name”:”warm”, “value”:0.1, “unit”:”PERCENTAGE”

}, {

“name”:”dry”, “value”:0, “unit”:”PERCENTAGE”

}

]

}, “sunshineDuration”:{

“value”:112, “unit”:”HOURS”

}, “hasAutoAssist”:true, “openWindowDetectionTimes”:5, “setbackScheduleDurationPerDay”:{

“value”:9.100000381469727, “unit”:”HOURS”

}, “totalSavingsInThermostaticModeAvailable”:false, “yearMonth”:”2022-09”, “hideOpenWindowDetection”:false, “home”:283787, “hideCommunityNews”:false

}

5.1.11 Heating circuits

class libtado.api.Tado.**get_heating_circuits**

Gets the heating circuits in the current home

Returns list of all dictionaries for all heating circuits

5.1.12 Home

class libtado.api.Tado.**get_home**
Get information about the home.

Returns A dictionary with information about your home.

Return type dict

Example

```
{
  'address': {
    'addressLine1': 'SOME_STREET',
    'addressLine2': None,
    'city': 'SOME_CITY',
    'country': 'SOME_COUNTRY',
    'state': None,
    'zipCode': 'SOME_ZIP_CODE'
  },
  'contactDetails': {
    'email': 'SOME_EMAIL',
    'name': 'SOME_NAME',
    'phone': 'SOME_PHONE'
  },
  'dateTimeZone': 'Europe/Berlin',
  'geolocation': {
    'latitude': SOME_LAT,
    'longitude': SOME_LONG
  },
  'id': SOME_ID,
  'installationCompleted': True,
  'name': 'SOME_NAME',
  'partner': None,
  'simpleSmartScheduleEnabled': True,
  'temperatureUnit': 'CELSIUS'
}
```

5.1.13 Home state

class libtado.api.Tado.**get_home_state**
Get information about the status of the home.

Returns A dictionary with the status of the home.

Return type dict

5.1.14 Incidents

class libtado.api.Tado.**get_incidents**
Gets the ongoing incidents in the current home

Returns Incident information

Return type dict

5.1.15 Installations

class libtado.api.Tado.**get_installations**
Gets the ongoing installations in the current home

Returns list of all current installations

5.1.16 Invitations

class libtado.api.Tado.**get_invitations**
Get active invitations.

Returns A list of active invitations to your home.

Return type list

Example

```
[
  {
    'email': 'SOME_INVITED_EMAIL',
    'firstSent': '2017-02-20T21:01:44.450Z',
    'home': {
      'address': {
        'addressLine1': 'SOME_STREET',
        'addressLine2': None,
        'city': 'SOME_CITY',
        'country': 'SOME_COUNTRY',
        'state': None,
        'zipCode': 'SOME_ZIP_CODE'
      },
      'contactDetails': {
        'email': 'SOME_EMAIL',
        'name': 'SOME_NAME',
        'phone': 'SOME_PHONE'
      },
      'dateTimeZone': 'Europe/Berlin',
      'geolocation': {
        'latitude': SOME_LAT,
        'longitude': SOME_LONG
      },
      'id': SOME_ID,
      'installationCompleted': True,
      'name': 'SOME_NAME',
      'partner': None,
      'simpleSmartScheduleEnabled': True,
      'temperatureUnit': 'CELSIUS'
    },
    'inviter': {
      'email': 'SOME_INVITER_EMAIL',
      'enabled': True,
      'homeId': SOME_ID,
      'locale': 'SOME_LOCALE',
      'name': 'SOME_NAME',
      'type': 'WEB_USER',
      'username': 'SOME_USERNAME'
```

(continues on next page)

(continued from previous page)

```
    },
    'lastSent': '2017-02-20T21:01:44.450Z',
    'token': 'SOME_TOKEN'
  }
]
```

5.1.17 Me

class libtado.api.Tado.**get_me**

Get information about the current user.

Returns A dictionary with information about the current user.

Return type dict

Example

```
{
  'email': 'SOME_EMAIL',
  'homes': [
    {
      'id': SOME_ID,
      'name': 'SOME_NAME'
    }
  ],
  'locale': 'en_US',
  'mobileDevices': [],
  'name': 'SOME_NAME',
  'username': 'SOME_USERNAME',
  'secret': 'SOME_CLIENT_SECRET'
}
```

5.1.18 Measuring device

class libtado.api.Tado.**get_measuring_device**

Gets the active measuring device of a zone

Parameters **zone** (*int*) – The zone ID.

Returns A dictionary with the current measuring informations.

Return type dict

5.1.19 Mobile devices

class libtado.api.Tado.**get_mobile_devices**

Get all mobile devices.

5.1.20 Open window detection

class libtado.api.Tado.**set_open_window_detection**

Get the open window detection for a zone

Parameters

- **zone** (*int*) – The zone ID.
- **enabled** (*bool*) – If open window detection is enabled
- **seconds** (*int*) – timeout in seconds

5.1.21 Report

class libtado.api.Tado.**get_report**

Parameters

- **zone** (*int*) – The zone ID.
- **date** (*str*) – The date in ISO8601 format. e.g. "2019-02-14"

Returns The daily report.

Return type dict

5.1.22 Schedule Timetables

class libtado.api.Tado.**get_schedule_timetables**

Gets the schedule timetables supported by the zone

Parameters **zone** (*int*) – The zone ID.

Returns The schedule types

Return type dict

5.1.23 Schedule

class libtado.api.Tado.**get_schedule**

Get the type of the currently configured schedule of a zone.

Parameters **zone** (*int*) – The zone ID.

Returns A dictionary with the ID and type of the schedule of the zone.

Return type dict

Tado allows three different types of a schedule for a zone:

- The same schedule for all seven days of a week.
- One schedule for weekdays, one for saturday and one for sunday.
- Seven different schedules - one for every day of the week.

Example

```
{
  'id': 1,
  'type': 'THREE_DAY'
}
```

5.1.24 Schedule blocks

class libtado.api.Tado.**get_schedule_blocks**

Gets the blocks for the current schedule on a zone

Parameters

- **zone** (*int*) – The zone ID.
- **schedule** (*int*) – The schedule ID to fetch

Returns The blocks for the requested schedule

Return type list

5.1.25 State

class libtado.api.Tado.**get_state**

Get the current state of a zone including its desired and current temperature. Check out the example output for more.

Parameters **zone** (*int*) – The zone ID.

Returns A dictionary with the current settings and sensor measurements of the zone.

Return type dict

Example

```
{
  'activityDataPoints': {
    'heatingPower': {
      'percentage': 0.0,
      'timestamp': '2017-02-21T11:56:52.204Z',
      'type': 'PERCENTAGE'
    }
  },
  'geolocationOverride': False,
  'geolocationOverrideDisableTime': None,
  'link': {'state': 'ONLINE'},
  'overlay': None,
  'overlayType': None,
  'preparation': None,
  'sensorDataPoints': {
    'humidity': {
      'percentage': 44.0,
      'timestamp': '2017-02-21T11:56:45.369Z',
      'type': 'PERCENTAGE'
    },
    'insideTemperature': {
      'celsius': 18.11,
      'fahrenheit': 64.6,
      'precision': {
        'celsius': 1.0,
        'fahrenheit': 1.0
      },
      'timestamp': '2017-02-21T11:56:45.369Z',
      'type': 'TEMPERATURE'
```

(continues on next page)

(continued from previous page)

```
    }
  },
  'setting': {
    'power': 'ON',
    'temperature': {
      'celsius': 20.0,
      'fahrenheit': 68.0
    },
    'type': 'HEATING'
  },
  'tadoMode': 'HOME'
}
```

5.1.26 Temperature offset

class libtado.api.Tado.**get_temperature_offset**

Gets the temperature offset of a device

Returns A dictionary that returns the offset in 'celsius' and 'fahrenheit'

Return type dict

Example

```
{
     "celsius": 0.0,
     "fahrenheit": 0.0
}
```

5.1.27 Users

class libtado.api.Tado.**get_users**

Get all users of your home.

5.1.28 Weather

class libtado.api.Tado.**get_weather**

Get the current weather of the location of your home.

Returns A dictionary with weather information for your home.

Return type dict

Example

```
{
  'outsideTemperature': {
    'celsius': 8.49,
    'fahrenheit': 47.28,
    'precision': {
```

(continues on next page)

(continued from previous page)

```
      'celsius': 0.01,
      'fahrenheit': 0.01
    },
    'timestamp': '2017-02-21T12:06:11.296Z',
    'type': 'TEMPERATURE'
  },
  'solarIntensity': {
    'percentage': 58.4,
    'timestamp': '2017-02-21T12:06:11.296Z',
    'type': 'PERCENTAGE'
  },
  'weatherState': {
    'timestamp': '2017-02-21T12:06:11.296Z',
    'type': 'WEATHER_STATE',
    'value': 'CLOUDY_PARTLY'
  }
}
```

5.1.29 Zones

class libtado.api.Tado.**get_zones**

Get all zones of your home.

Returns A list of dictionaries with all your zones.

Return type list

Example

```
[
  { 'dateCreated': '2016-12-23T15:53:43.615Z',
    'dazzleEnabled': True,
    'deviceTypes': ['VA01'],
    'devices': [
      {
        'characteristics': {
          'capabilities': [ 'INSIDE_TEMPERATURE_MEASUREMENT', 'IDENTIFY']
        },
        'connectionState': {
          'timestamp': '2017-02-21T14:22:45.913Z',
          'value': True
        },
        'currentFwVersion': '36.15',
        'deviceType': 'VA01',
        'duties': ['ZONE_UI', 'ZONE_DRIVER', 'ZONE_LEADER'],
        'mountingState': {
          'timestamp': '2017-02-12T13:34:35.288Z',
          'value': 'CALIBRATED'
        },
        'serialNo': 'SOME_SERIAL',
        'shortSerialNo': 'SOME_SERIAL'
      }
    ],
    'id': 1,
```

(continues on next page)

(continued from previous page)


```
    'name': 'SOME_NAME',
    'reportAvailable': False,
    'supportsDazzle': True,
    'type': 'HEATING'
  },
  {
    'dateCreated': '2016-12-23T16:16:11.390Z',
    'dazzleEnabled': True,
    'deviceTypes': ['VA01'],
    'devices': [
      {
        'characteristics': {
          'capabilities': [ 'INSIDE_TEMPERATURE_MEASUREMENT', 'IDENTIFY']
        },
        'connectionState': {
          'timestamp': '2017-02-21T14:19:40.215Z',
          'value': True
        },
        'currentFwVersion': '36.15',
        'deviceType': 'VA01',
        'duties': ['ZONE_UI', 'ZONE_DRIVER', 'ZONE_LEADER'],
        'mountingState': {
          'timestamp': '2017-01-12T13:22:11.618Z',
          'value': 'CALIBRATED'
        },
        'serialNo': 'SOME_SERIAL',
        'shortSerialNo': 'SOME_SERIAL'
      }
    ],
    'id': 3,
    'name': 'SOME_NAME ',
    'reportAvailable': False,
    'supportsDazzle': True,
    'type': 'HEATING'
  }
]
```


5.2 Setter

5.2.1 Early start

class libtado.api.Tado.**set_early_start**

Enable or disable the early start feature of a zone.

Parameters

- **zone** (*int*) – The zone ID.
- **enabled** (*bool*) – Enable (True) or disable (False) the early start feature of the zone.

Returns The new configuration of the early start feature.

Return type dict

Example

```
{'enabled': True}
```

5.2.2 End manual control

class libtado.api.Tado.**end_manual_control**
End the manual control of a zone.

5.2.3 Home state

class libtado.api.Tado.**set_home_state**
Set at-home/away state

Parameters **at_home** (*bool*) – True for at HOME, false for AWAY.

5.2.4 Schedule

class libtado.api.Tado.**set_schedule**
Set the type of the currently configured schedule of a zone.

Parameters

- **zone** (*int*) – The zone ID.
- **schedule** (*int*) – The schedule to activate. The supported zones are currently
  - 0: ONE_DAY
  - 1: THREE_DAY
  - 2: SEVEN_DAY

  But the actual mapping should be retrieved via get_schedule_timetables.

Returns The new configuration

Return type dict

5.2.5 Running times

class libtado.api.Tado.**get_running_times**
Get all running times of your home.

Parameters **None.** –

Returns A dict of your running times.

Return type list

Example

```
{
```

“runningTimes”:[

{ “runningTimeInSeconds”:0, “startTime”:”2022-08-18 00:00:00”, “endTime”:”2022-08-19 00:00:00”, “zones”:[
{ “id”:1, “runningTimeInSeconds”:0
}, {
“id”:6, “runningTimeInSeconds”:0
}, {
“id”:11, “runningTimeInSeconds”:0
}, {
“id”:12, “runningTimeInSeconds”:0
}
]
}
], “summary”:{
“startTime”:”2022-08-18 00:00:00”, “endTime”:”2022-08-19 00:00:00”, “totalRunningTimeInSeconds”:0
}, “lastUpdated”:”2022-08-18T05:07:44Z”
}

5.2.6 Schedule blocks

class libtado.api.Tado.**set_schedule_blocks**

Sets the blocks for the current schedule on a zone

Parameters

- **zone** (*int*) – The zone ID.
- **schedule** (*int*) – The schedule ID.
- **blocks** (*list*) – The new blocks

Returns The new configuration

Return type list

5.2.7 Temperature

class libtado.api.Tado.**set_temperature**

Set the desired temperature of a zone.

Parameters

- **zone** (*int*) – The zone ID.
- **temperature** (*float*) – The desired temperature in celsius.
- **termination** (*str/int*) – The termination mode for the zone.

Returns A dictionary with the new zone settings.

Return type dict

If you set a desired temperature less than 5 celsius it will turn of the zone!

The termination supports three different mode:

- "MANUAL": The zone will be set on the desired temperature until you change it manually.
- "AUTO": The zone will be set on the desired temperature until the next automatic change.
- INTEGER: The zone will be set on the desired temperature for INTEGER seconds.

Example

```
{
  'setting': {
    'power': 'ON',
    'temperature': {'celsius': 12.0, 'fahrenheit': 53.6},
    'type': 'HEATING'
  },
  'termination': {
    'projectedExpiry': None,
    'type': 'MANUAL'
  },
  'type': 'MANUAL'
}
```

5.2.8 Temperature offset

class libtado.api.Tado.**set_temperature_offset**

Sets the temperature offset of a device

Parameters

- **device_serial** (*Str*) – The serial number of the device
- **offset** (*float*) – the temperature offset to apply in celsius

Returns A dictionary that returns the offset in 'celsius' and 'fahrenheit'

Return type dict

5.2.9 Zone name

class libtado.api.Tado.**set_zone_name**

Sets the name of the zone

Parameters

- **zone** (*int*) – The zone ID.
- **new_name** (*str*) – The new name of the zone

Returns dict

5.2.10 Zone states

class libtado.api.Tado.**get_zone_states**

Get all zone states of your home.

Parameters `None.` –

Returns A dict of your zone states.

Return type list

Example


```
{
```

“zoneStates”:{

“1”:{ “tadoMode”:”HOME”, “geolocationOverride”:false, “geolocationOverrideDisableTime”:”None”, “preparation”:”None”, “setting”:{

“type”:”HEATING”, “power”:”ON”, “temperature”:{

“celsius”:19.0, “fahrenheit”:66.2

}

}, “overlayType”:”None”, “overlay”:”None”, “openWindow”:”None”, “nextScheduleChange”:{

“start”:”2022-08-18T16:00:00Z”, “setting”:{

“type”:”HEATING”, “power”:”ON”, “temperature”:{

“celsius”:20.0, “fahrenheit”:68.0

}

}

}, “nextTimeBlock”:{

“start”:”2022-08-18T16:00:00.000Z”

}, “link”:{

“state”:”ONLINE”

}, “activityDataPoints”:{

“heatingPower”:{ “type”:”PERCENTAGE”, “percentage”:0.0, “timestamp”:”2022-08-18T05:34:32.127Z”

}

}, “sensorDataPoints”:{

“insideTemperature”:{ “celsius”:24.13, “fahrenheit”:75.43, “timestamp”:”2022-08-18T05:36:21.241Z”, “type”:”TEMPERATURE”, “precision”:{

“celsius”:0.1, “fahrenheit”:0.1

}

}, “humidity”:{

“type”:”PERCENTAGE”, “percentage”:62.2, “timestamp”:”2022-08-18T05:36:21.241Z”

}

}

}, “6”:{

“tadoMode”:”HOME”, “geolocationOverride”:false, “geolocationOverrideDisable-Time”:”None”, “preparation”:”None”, “setting”:{

“type”:”HEATING”, “power”:”ON”, “temperature”:{

“celsius”:19.5, “fahrenheit”:67.1

}

}, “overlayType”:”None”, “overlay”:”None”, “openWindow”:”None”, “nextSched-uleChange”:{

“start”:”2022-08-18T07:00:00Z”, “setting”:{

“type”:”HEATING”, “power”:”ON”, “temperature”:{

“celsius”:18.0, “fahrenheit”:64.4

}

}

}, “nextTimeBlock”:{

“start”:”2022-08-18T07:00:00.000Z”

}, “link”:{

“state”:”ONLINE”

}, “activityDataPoints”:{

“heatingPower”:{ “type”:”PERCENTAGE”, “percentage”:0.0, “timestamp”:”2022-08-18T05:47:58.505Z”

}

}, “sensorDataPoints”:{

“insideTemperature”:{ “celsius”:24.2, “fahrenheit”:75.56, “timestamp”:”2022-08-18T05:46:09.620Z”, “type”:”TEMPERATURE”, “precision”:{

“celsius”:0.1, “fahrenheit”:0.1

}

}, “humidity”:{

“type”:”PERCENTAGE”, “percentage”:64.8, “timestamp”:”2022-08-18T05:46:09.620Z”

}

}

}

}

CHAPTER 6

Step-by-Step Guide

You can download the library sources at https://github.com/germainlefebvre4/libtado:

6.1 Requirements

The library development requires at least python `3.7`.

This library is tested with following python versions:

- `3.7`
- `3.8`

6.2 Setup

Update your system and install a python version (at least the minimum required) and install the python virtualenv tol *pipenv*.

```
sudo apt update
sudo apt install python3.7 python3.7-pip
sudo pip install pipenv
```

Initialize your *pipenv* setup and install all the development libraries

```
pipenv update --dev
```

6.3 Improve the library

The library is served through 2 sections:

- API in `./libtado/api.py`
- CLI in `./libtado/cli.py`

6.4 Write and run some tests

Unit tests are important for the developer team because it add strenghtness and confidence to the code.

The tests are written in the following files:

- Global Tado in `./tests/global/test_tado.py`
- Library API in `./tests/api/test_api.py`
- Library CLI in `./tests/cli/test_cli.py`

Run the tests inside pipenv.

```
pipenv run pytest tests/
```

CHAPTER 7

License

CHAPTER 8

References

Index