Homematic IP Rest API Documentation

Release 1.0.4

Heimo Stieg

Jul 12, 2022

Contents

This documentation is for a **Python 3** wrapper for the homematicIP REST API (Access Point Based) Since there is no official documentation about this API everything was done via reverse engineering. Use at your own risk.

CHAPTER 1

Getting Started

1.1 Installation

Just run **pip3 install -U homematicip** in the command line to get the package. This will install (and update) the library and all required packages

1.2 Getting the AUTH-TOKEN

Before you can start using the library you will need an auth-token. Otherwise the HMIP Cloud will not trust you.

You will need:

- Access to an active Access Point (it must glow blue)
- the SGTIN of the Access Point
- [optional] the PIN

Now you have to run **hmip_generate_auth_token.py** and follow it's instructions. It will generate a **config.ini** in your current working directory. The scripts which are using this library are looking for this file to load the auth-token and SGTIN of the Access Point. You can either place it in the working directory when you are running the scripts or depending on your OS in different "global" folders:

- General
  - current working directory
- Windows
  - %APPDATA%\homematicip-rest-api
  - %PROGRAMDATA%\homematicip-rest-api
- Linux
  - ~/.homematicip-rest-api/

  - /etc/homematicip-rest-api/
- MAC OS
  - ~/Library/Preferences/homematicip-rest-api/
  - /Library/Application Support/homematicip-rest-api/

CHAPTER 2

Homematic IP Overview

2.1 General

The library structure is similar to the REST API or HomematicIP Android/IOS App. The library has two ways of communicating with the REST API. Either via requests (homematicip package) or via async calls (homematicip.aio package).

2.2 Important terms

- Home: is the most important object as it has the "overview" of the installation
- Device: a hardware device e.g. shutter contact, heating thermostat, alarm siren, ...
- Group: a group of devices for a specific need. E.g. Heating group, security group, ...
- MetaGroup: a collection of groups. In the HomematicIP App this is called a "Room"

CHAPTER 3

homematicip

3.1 homematicip package

3.1.1 Subpackages

homematicip.aio package

Submodules

homematicip.aio.auth module

class homematicip.aio.auth.**AsyncAuth**(*loop*, *websession=None*)
Bases: *homematicip.auth.Auth*

this class represents the 'Async Auth' of the homematic ip

confirmAuthToken(*authToken*)

connectionRequest(*devicename='homematicip-async'*)

init(*access_point_id*, *lookup=True*, *lookup_url=None*)

isRequestAcknowledged()

requestAuthToken()

class homematicip.aio.auth.**AsyncAuthConnection**(*loop*, *session=None*)
Bases: *homematicip.aio.connection.AsyncConnection*

homematicip.aio.class_maps module

homematicip.aio.connection module

class homematicip.aio.connection.**AsyncConnection**(*loop*, *session=None*)
Bases: *homematicip.base.base_connection.BaseConnection*

Handles async http and websocket traffic.

api_call(*path*, *body=None*, *full_url=False*)
Make the actual call to the HMIP server.

Throws *HmipWrongHttpStatusError* or *HmipConnectionError* if connection has failed or response is not correct.

close_websocket_connection(*source_is_reading_loop=False*)

connect_timeout = 20

full_url(*partial_url*)

init(*accesspoint_id*, *lookup=True*, *lookup_url='https://lookup.homematic.com:48335/getHost'*, ***kwargs*)

ping_loop = 60

ping_timeout = 3

ws_connect(***, *on_message*, *on_error*)

ws_connected
Websocket is connected.

homematicip.aio.device module

class homematicip.aio.device.**AsyncAccelerationSensor**(*connection*)
Bases: *homematicip.device.AccelerationSensor*, *homematicip.aio.device.AsyncDevice*

HMIP-SAM

set_acceleration_sensor_event_filter_period(*period: float*, *channelIndex=1*)

set_acceleration_sensor_mode(*mode: homematicip.base.enums.AccelerationSensorMode*, *channelIndex=1*)

set_acceleration_sensor_neutral_position(*neutralPosition: homematicip.base.enums.AccelerationSensorNeutralPosition*, *channelIndex=1*)

set_acceleration_sensor_sensitivity(*sensitivity: homematicip.base.enums.AccelerationSensorSensitivity*, *channelIndex=1*)

set_acceleration_sensor_trigger_angle(*angle: int*, *channelIndex=1*)

set_notification_sound_type(*soundType: homematicip.base.enums.NotificationSoundType*, *isHighToLow: bool*, *channelIndex=1*)

class homematicip.aio.device.**AsyncAlarmSirenIndoor**(*connection*)
Bases: *homematicip.device.AlarmSirenIndoor*, *homematicip.aio.device.AsyncSabotageDevice*

HMIP-ASIR (Alarm Siren)

class homematicip.aio.device.**AsyncAlarmSirenOutdoor**(*connection*)
Bases: *homematicip.device.AlarmSirenOutdoor*, *homematicip.aio.device.AsyncAlarmSirenIndoor*

HMIP-ASIR-O (Alarm Siren Outdoor)

class homematicip.aio.device.**AsyncBlind**(*connection*)
Bases: *homematicip.device.Blind*, *homematicip.aio.device.AsyncShutter*

Base class for async blind devices

set_slats_level(*slatsLevel=0.0*, *shutterLevel=None*, *channelIndex=1*)
sets the slats and shutter level

Parameters

- **slatsLevel** (*float*) – the new level of the slats. 0.0 = open, 1.0 = closed,
- **shutterLevel** (*float*) – the new level of the shutter. 0.0 = open, 1.0 = closed, None = use the current value
- **channelIndex** (*int*) – the channel to control

Returns the result of the _restCall

class homematicip.aio.device.**AsyncBlindModule**(*connection*)
Bases: *homematicip.device.BlindModule*, *homematicip.aio.device.AsyncDevice*

HMIP-HDM1 (Hunter Douglas & erfal window blinds)

set_primary_shading_level(*primaryShadingLevel: float*)

set_secondary_shading_level(*primaryShadingLevel: float*, *secondaryShadingLevel: float*)

stop()
stops the current operation :returns: the result of the _restCall

class homematicip.aio.device.**AsyncBrandBlind**(*connection*)
Bases: *homematicip.device.BrandBlind*, *homematicip.aio.device.AsyncFullFlushBlind*

HMIP-BBL (Blind Actuator for brand switches)

class homematicip.aio.device.**AsyncBrandDimmer**(*connection*)
Bases: *homematicip.aio.device.AsyncDimmer*

HMIP-BDT Brand Dimmer

class homematicip.aio.device.**AsyncBrandPushButton**(*connection*)
Bases: *homematicip.device.BrandPushButton*, *homematicip.aio.device.AsyncPushButton*

HMIP-BRC2 (Remote Control for brand switches – 2x channels)

class homematicip.aio.device.**AsyncBrandSwitchMeasuring**(*connection*)
Bases: *homematicip.device.BrandSwitchMeasuring*, *homematicip.aio.device.AsyncSwitchMeasuring*

HMIP-BSM (Brand Switch and Meter)

class homematicip.aio.device.**AsyncBrandSwitchNotificationLight**(*connection*)
Bases: *homematicip.device.BrandSwitchNotificationLight*, *homematicip.aio.device.AsyncSwitch*

HMIP-BSL (Switch Actuator for brand switches – with signal lamp)

set_rgb_dim_level(*channelIndex: int*, *rgb: homematicip.base.enums.RGBColorState*, *dimLevel: float*)
sets the color and dimlevel of the lamp

Parameters

- **channelIndex** (*int*) – the channelIndex of the lamp. Use self.topLightChannelIndex or self.bottomLightChannelIndex
- **rgb** (RGBColorState) – the color of the lamp
- **dimLevel** (*float*) – the dimLevel of the lamp. 0.0 = off, 1.0 = MAX

Returns the result of the _restCall

set_rgb_dim_level_with_time(*channelIndex: int*, *rgb: homematicip.base.enums.RGBColorState*, *dimLevel: float*, *onTime: float*, *rampTime: float*)
sets the color and dimlevel of the lamp

Parameters

- **channelIndex** (*int*) – the channelIndex of the lamp. Use self.topLightChannelIndex or self.bottomLightChannelIndex
- **rgb** (RGBColorState) – the color of the lamp
- **dimLevel** (*float*) – the dimLevel of the lamp. 0.0 = off, 1.0 = MAX
- **onTime** (*float*) –
- **rampTime** (*float*) –

Returns the result of the _restCall

class homematicip.aio.device.**AsyncContactInterface**(*connection*)
Bases: *homematicip.device.ContactInterface*, *homematicip.aio.device.AsyncSabotageDevice*

HMIP-SCI (Contact Interface Sensor)

class homematicip.aio.device.**AsyncDevice**(*connection*)
Bases: *homematicip.device.Device*

Async implementation of a genereric homematic ip device

authorizeUpdate()

delete()

is_update_applicable()

set_label(*label*)

set_router_module_enabled(*enabled=True*)

class homematicip.aio.device.**AsyncDimmer**(*connection*)
Bases: *homematicip.device.Dimmer*, *homematicip.aio.device.AsyncDevice*

Base dimmer device class

set_dim_level(*dimLevel=0.0*, *channelIndex=1*)

class homematicip.aio.device.**AsyncDinRailBlind4**(*connection*)
Bases: *homematicip.device.DinRailBlind4*, *homematicip.aio.device.AsyncBlind*

HmIP-DRBLI4 (Blind Actuator for DIN rail mount – 4 channels)

class homematicip.aio.device.**AsyncDinRailSwitch**(*connection*)
Bases: *homematicip.device.DinRailSwitch*, *homematicip.aio.device.AsyncFullFlushInputSwitch*

HMIP-DRSI1 (Switch Actuator for DIN rail mount – 1x channel)

class homematicip.aio.device.**AsyncDinRailSwitch4**(*connection*)
Bases: *homematicip.device.DinRailSwitch4*, *homematicip.aio.device.AsyncSwitch*

HMIP-DRSI4 (Homematic IP Switch Actuator for DIN rail mount – 4x channels)

class homematicip.aio.device.**AsyncDoorModule**(*connection*)
Bases: *homematicip.device.DoorModule*, *homematicip.aio.device.AsyncDevice*

Generic Door Module class

send_door_command(*doorCommand=<DoorCommand.STOP: 'STOP'>*)

class homematicip.aio.device.**AsyncFloorTerminalBlock10**(*connection*)
Bases: *homematicip.device.FloorTerminalBlock10*, *homematicip.aio.device.AsyncFloorTerminalBlock6*

HMIP-FAL24-C10 (Floor Heating Actuator – 10x channels, 24V)

class homematicip.aio.device.**AsyncFloorTerminalBlock12**(*connection*)
Bases: *homematicip.device.FloorTerminalBlock12*, *homematicip.aio.device.AsyncDevice*

HMIP-FALMOT-C12 (Floor Heating Actuator – 12x channels, motorised)

set_minimum_floor_heating_valve_position(*minimumFloorHeatingValvePosition: float*)
sets the minimum floot heating valve position

Parameters **minimumFloorHeatingValvePosition** (*float*) – the minimum valve position. must be between 0.0 and 1.0

Returns the result of the _restCall

class homematicip.aio.device.**AsyncFloorTerminalBlock6**(*connection*)
Bases: *homematicip.device.FloorTerminalBlock6*, *homematicip.aio.device.AsyncDevice*

HMIP-FAL230-C6 (Floor Heating Actuator - 6 channels, 230 V)

class homematicip.aio.device.**AsyncFullFlushBlind**(*connection*)
Bases: *homematicip.device.FullFlushBlind*, *homematicip.aio.device.AsyncBlind*

HMIP-FBL (Blind Actuator - flush-mount)

class homematicip.aio.device.**AsyncFullFlushContactInterface**(*connection*)
Bases: *homematicip.device.FullFlushContactInterface*, *homematicip.aio.device.AsyncDevice*

HMIP-FCI1 (Contact Interface flush-mount – 1 channel)

class homematicip.aio.device.**AsyncFullFlushContactInterface6**(*connection*)
Bases: *homematicip.device.FullFlushContactInterface6*, *homematicip.aio.device.AsyncDevice*

HMIP-FCI6 (Contact Interface flush-mount – 6 channels)

class homematicip.aio.device.**AsyncFullFlushDimmer**(*connection*)
Bases: *homematicip.aio.device.AsyncDimmer*

HMIP-FDT Dimming Actuator flush-mount

class homematicip.aio.device.**AsyncFullFlushInputSwitch**(*connection*)
Bases: *homematicip.device.FullFlushInputSwitch*, *homematicip.aio.device.AsyncSwitch*

HMIP-FSI16 (Switch Actuator with Push-button Input 230V, 16A)

class homematicip.aio.device.**AsyncFullFlushShutter**(*connection*)
Bases: *homematicip.device.FullFlushShutter*, *homematicip.aio.device.AsyncShutter*

HMIP-FROLL (Shutter Actuator - flush-mount) / HMIP-BROLL (Shutter Actuator - Brand-mount)

class homematicip.aio.device.**AsyncFullFlushSwitchMeasuring**(*connection*)
Bases: *homematicip.device.FullFlushSwitchMeasuring*, *homematicip.aio.device.AsyncSwitchMeasuring*

HMIP-FSM (Full flush Switch and Meter)

class homematicip.aio.device.**AsyncGarageDoorModuleTormatic**(*connection*)
Bases: *homematicip.device.GarageDoorModuleTormatic*, *homematicip.aio.device.AsyncDoorModule*

HMIP-MOD-TM (Garage Door Module Tormatic)

class homematicip.aio.device.**AsyncHeatingSwitch2**(*connection*)
Bases: *homematicip.device.HeatingSwitch2*, *homematicip.aio.device.AsyncSwitch*

HMIP-WHS2 (Switch Actuator for heating systems – 2x channels)

class homematicip.aio.device.**AsyncHeatingThermostat**(*connection*)
Bases: *homematicip.device.HeatingThermostat*, *homematicip.aio.device.AsyncOperationLockableDevice*

HMIP-eTRV (Radiator Thermostat)

class homematicip.aio.device.**AsyncHeatingThermostatCompact**(*connection*)
Bases: *homematicip.device.HeatingThermostatCompact*, *homematicip.aio.device.AsyncSabotageDevice*

HMIP-eTRV-C (Heating-thermostat compact without display)

class homematicip.aio.device.**AsyncHeatingThermostatEvo**(*connection*)
Bases: *homematicip.device.HeatingThermostatEvo*, *homematicip.aio.device.AsyncSabotageDevice*

HMIP-eTRV-E (Heating-thermostat new evo version)

class homematicip.aio.device.**AsyncHoermannDrivesModule**(*connection*)
Bases: *homematicip.device.HoermannDrivesModule*, *homematicip.aio.device.AsyncDoorModule*

HMIP-MOD-HO (Garage Door Module for Hörmann)

class homematicip.aio.device.**AsyncHomeControlAccessPoint**(*connection*)
Bases: *homematicip.device.HomeControlAccessPoint*, *homematicip.aio.device.AsyncDevice*

HMIP-HAP

class homematicip.aio.device.**AsyncKeyRemoteControl4**(*connection*)
Bases: *homematicip.device.KeyRemoteControl4*, *homematicip.aio.device.AsyncPushButton*

HMIP-KRC4 (Key Ring Remote Control - 4 buttons)

class homematicip.aio.device.**AsyncKeyRemoteControlAlarm**(*connection*)
Bases: *homematicip.device.KeyRemoteControlAlarm*, *homematicip.aio.device.AsyncDevice*

HMIP-KRCA (Key Ring Remote Control - alarm)

class homematicip.aio.device.**AsyncLightSensor**(*connection*)
Bases: *homematicip.device.LightSensor*, *homematicip.aio.device.AsyncDevice*

Async implementation of HMIP-SLO (Light Sensor outdoor)

class homematicip.aio.device.**AsyncMotionDetectorIndoor**(*connection*)
Bases: *homematicip.device.MotionDetectorIndoor*, *homematicip.aio.device.AsyncSabotageDevice*

HMIP-SMI (Motion Detector with Brightness Sensor - indoor)

class homematicip.aio.device.**AsyncMotionDetectorOutdoor**(*connection*)
Bases: *homematicip.device.MotionDetectorOutdoor*, *homematicip.aio.device.AsyncDevice*

HMIP-SMO-A (Motion Detector with Brightness Sensor - outdoor)

class homematicip.aio.device.**AsyncMotionDetectorPushButton**(*connection*)
Bases: *homematicip.device.MotionDetectorPushButton*, *homematicip.aio.device.AsyncDevice*

HMIP-SMI55 (Motion Detector with Brightness Sensor and Remote Control - 2-button)

class homematicip.aio.device.**AsyncMultiIOBox**(*connection*)
Bases: *homematicip.device.MultiIOBox*, *homematicip.aio.device.AsyncSwitch*

HMIP-MIOB (Multi IO Box for floor heating & cooling)

class homematicip.aio.device.**AsyncOpenCollector8Module**(*connection*)
Bases: *homematicip.device.OpenCollector8Module*, *homematicip.aio.device.AsyncSwitch*

Async implementation of HMIP-MOD-OC8 (Open Collector Module)

class homematicip.aio.device.**AsyncOperationLockableDevice**(*connection*)
Bases: *homematicip.device.OperationLockableDevice*, *homematicip.aio.device.AsyncDevice*

set_operation_lock(*operationLock=True*)

class homematicip.aio.device.**AsyncPassageDetector**(*connection*)
Bases: *homematicip.device.PassageDetector*, *homematicip.aio.device.AsyncSabotageDevice*

HMIP-SPDR (Passage Detector)

class homematicip.aio.device.**AsyncPlugableSwitch**(*connection*)
Bases: *homematicip.device.PlugableSwitch*, *homematicip.aio.device.AsyncSwitch*

Async implementation of HMIP-PS (Pluggable Switch)

class homematicip.aio.device.**AsyncPlugableSwitchMeasuring**(*connection*)
Bases: *homematicip.device.PlugableSwitchMeasuring*, *homematicip.aio.device.AsyncSwitchMeasuring*

HMIP-PSM (Pluggable Switch and Meter)

class homematicip.aio.device.**AsyncPluggableDimmer**(*connection*)
Bases: *homematicip.aio.device.AsyncDimmer*

HMIP-PDT Pluggable Dimmer

class homematicip.aio.device.**AsyncPluggableMainsFailureSurveillance**(*connection*)
Bases: *homematicip.device.PluggableMainsFailureSurveillance*, *homematicip.aio.device.AsyncDevice*

[HMIP-PMFS] (Plugable Power Supply Monitoring)

class homematicip.aio.device.**AsyncPresenceDetectorIndoor**(*connection*)
Bases: *homematicip.device.PresenceDetectorIndoor*, *homematicip.aio.device.AsyncSabotageDevice*

HMIP-SPI (Presence Sensor - indoor)

class homematicip.aio.device.**AsyncPrintedCircuitBoardSwitch2**(*connection*)
Bases: *homematicip.device.PrintedCircuitBoardSwitch2*, *homematicip.aio.device.AsyncSwitch*

Async implementation of HMIP-PCBS2 (Switch Circuit Board - 2x channels)

class homematicip.aio.device.**AsyncPrintedCircuitBoardSwitchBattery**(*connection*)
Bases: *homematicip.device.PrintedCircuitBoardSwitchBattery*, *homematicip.aio.device.AsyncSwitch*

HMIP-PCBS-BAT (Printed Circuit Board Switch Battery)

class homematicip.aio.device.**AsyncPushButton**(*connection*)
Bases: *homematicip.device.PushButton*, *homematicip.aio.device.AsyncDevice*

HMIP-WRC2 (Wall-mount Remote Control - 2-button)

class homematicip.aio.device.**AsyncPushButton6**(*connection*)
Bases: *homematicip.device.PushButton6*, *homematicip.aio.device.AsyncPushButton*

HMIP-WRC6 (Wall-mount Remote Control - 6-button)

class homematicip.aio.device.**AsyncPushButtonFlat**(*connection*)
Bases: *homematicip.device.PushButtonFlat*, *homematicip.aio.device.AsyncPushButton*

HMIP-WRCC2 (Wall-mount Remote Control – flat)

class homematicip.aio.device.**AsyncRainSensor**(*connection*)
Bases: *homematicip.device.RainSensor*, *homematicip.aio.device.AsyncDevice*

HMIP-SRD (Rain Sensor)

class homematicip.aio.device.**AsyncRemoteControl8**(*connection*)
Bases: *homematicip.device.RemoteControl8*, *homematicip.aio.device.AsyncPushButton*

HMIP-RC8 (Remote Control - 8 buttons)

class homematicip.aio.device.**AsyncRemoteControl8Module**(*connection*)
Bases: *homematicip.device.RemoteControl8Module*, *homematicip.aio.device.AsyncRemoteControl8*

HMIP-MOD-RC8 (Open Collector Module Sender - 8x)

class homematicip.aio.device.**AsyncRoomControlDevice**(*connection*)
Bases: *homematicip.device.RoomControlDevice*, *homematicip.aio.device.AsyncWallMountedThermostatPro*

ALPHA-IP-RBG (Alpha IP Wall Thermostat Display)

class homematicip.aio.device.**AsyncRoomControlDeviceAnalog**(*connection*)
Bases: *homematicip.aio.device.AsyncDevice*

ALPHA-IP-RBGa (ALpha IP Wall Thermostat Display analog)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.aio.device.**AsyncRotaryHandleSensor**(*connection*)
Bases: *homematicip.device.RotaryHandleSensor*, *homematicip.aio.device.AsyncSabotageDevice*

HMIP-SRH

class homematicip.aio.device.**AsyncSabotageDevice**(*connection*)
Bases: *homematicip.device.SabotageDevice*, *homematicip.aio.device.AsyncDevice*

Async implementation sabotage signaling devices

class homematicip.aio.device.**AsyncShutter**(*connection*)
Bases: *homematicip.device.Shutter*, *homematicip.aio.device.AsyncDevice*

Base class for async shutter devices

set_shutter_level(*level=0.0*, *channelIndex=1*)
sets the shutter level

Parameters

- **level** (*float*) – the new level of the shutter. 0.0 = open, 1.0 = closed
- **channelIndex** (*int*) – the channel to control

Returns the result of the _restCall

set_shutter_stop(*channelIndex=1*)
stops the current shutter operation

Parameters **channelIndex** (*int*) – the channel to control

Returns the result of the _restCall

class homematicip.aio.device.**AsyncShutterContact**(*connection*)
Bases: *homematicip.device.ShutterContact*, *homematicip.aio.device.AsyncSabotageDevice*

HMIP-SWDO (Door / Window Contact - optical) / HMIP-SWDO-I (Door / Window Contact Invisible - optical)

class homematicip.aio.device.**AsyncShutterContactMagnetic**(*connection*)
Bases: *homematicip.device.ShutterContactMagnetic*, *homematicip.aio.device.AsyncDevice*

HMIP-SWDM / HMIP-SWDM-B2 (Door / Window Contact - magnetic

class homematicip.aio.device.**AsyncShutterContactOpticalPlus**(*connection*)
Bases: *homematicip.device.ShutterContactOpticalPlus*, *homematicip.aio.device.AsyncShutterContact*

HmIP-SWDO-PL (Window / Door Contact – optical, plus)

class homematicip.aio.device.**AsyncSmokeDetector**(*connection*)
Bases: *homematicip.device.SmokeDetector*, *homematicip.aio.device.AsyncDevice*

HMIP-SWSD (Smoke Alarm with Q label)

class homematicip.aio.device.**AsyncSwitch**(*connection*)
Bases: *homematicip.device.Switch*, *homematicip.aio.device.AsyncDevice*

Generic async switch

set_switch_state(*on=True*, *channelIndex=1*)

turn_off(*channelIndex=1*)

turn_on(*channelIndex=1*)

class homematicip.aio.device.**AsyncSwitchMeasuring**(*connection*)
Bases: *homematicip.device.SwitchMeasuring*, *homematicip.aio.device.AsyncSwitch*

Generic async switch measuring

reset_energy_counter()

class homematicip.aio.device.**AsyncTemperaturDifferenceSensor2**(*connection*)
Bases: *homematicip.device.TemperaturDifferenceSensor2*, *homematicip.aio.device.AsyncDevice*

HmIP-STE2-PCB (Temperature Difference Sensors - 2x sensors)

class homematicip.aio.device.**AsyncTemperatureHumiditySensorDisplay**(*connection*)
Bases: *homematicip.device.TemperatureHumiditySensorDisplay*, *homematicip.aio.device.AsyncDevice*

HMIP-STHD (Temperature and Humidity Sensor with display - indoor)

set_display(*display: homematicip.base.enums.ClimateControlDisplay = <ClimateControlDisplay.ACTUAL: 'ACTUAL'>*)

class homematicip.aio.device.**AsyncTemperatureHumiditySensorOutdoor**(*connection*)
Bases: *homematicip.device.TemperatureHumiditySensorOutdoor*, *homematicip.aio.device.AsyncDevice*

HMIP-STHO (Temperature and Humidity Sensor outdoor)

class homematicip.aio.device.**AsyncTemperatureHumiditySensorWithoutDisplay**(*connection*)
Bases: *homematicip.device.TemperatureHumiditySensorWithoutDisplay*, *homematicip.aio.device.AsyncDevice*

HMIP-STH (Temperature and Humidity Sensor without display - indoor)

class homematicip.aio.device.**AsyncTiltVibrationSensor**(*connection*)
Bases: *homematicip.device.TiltVibrationSensor*, *homematicip.aio.device.AsyncDevice*

HMIP-STV (Inclination and vibration Sensor)

set_acceleration_sensor_event_filter_period(*period: float*, *channelIndex=1*)

set_acceleration_sensor_mode(*mode: homematicip.base.enums.AccelerationSensorMode*, *channelIndex=1*)

set_acceleration_sensor_sensitivity(*sensitivity: homematicip.base.enums.AccelerationSensorSensitivity*, *channelIndex=1*)

set_acceleration_sensor_trigger_angle(*angle: int*, *channelIndex=1*)

class homematicip.aio.device.**AsyncWallMountedThermostatBasicHumidity**(*connection*)
Bases: *homematicip.aio.device.AsyncWallMountedThermostatPro*

HMIP-WTH-B (Wall Thermostat – basic)

class homematicip.aio.device.**AsyncWallMountedThermostatPro**(*connection*)
Bases: *homematicip.device.WallMountedThermostatPro*, *homematicip.aio.device.AsyncTemperatureHumiditySensorDisplay*, *homematicip.aio.device.AsyncOperationLockableDevice*

HMIP-WTH, HMIP-WTH-2 (Wall Thermostat with Humidity Sensor) / HMIP-BWTH (Brand Wall Thermostat with Humidity Sensor)

class homematicip.aio.device.**AsyncWaterSensor**(*connection*)
Bases: *homematicip.device.WaterSensor*, *homematicip.aio.device.AsyncDevice*

HMIP-SWD

set_acoustic_alarm_signal(*acousticAlarmSignal: homematicip.base.enums.AcousticAlarmSignal*)

set_acoustic_alarm_timing(*acousticAlarmTiming: homematicip.base.enums.AcousticAlarmTiming*)

set_acoustic_water_alarm_trigger(*acousticWaterAlarmTrigger: homematicip.base.enums.WaterAlarmTrigger*)

set_inapp_water_alarm_trigger(*inAppWaterAlarmTrigger: homematicip.base.enums.WaterAlarmTrigger*)

set_siren_water_alarm_trigger(*sirenWaterAlarmTrigger: homematicip.base.enums.WaterAlarmTrigger*)

class homematicip.aio.device.**AsyncWeatherSensor**(*connection*)
Bases: *homematicip.device.WeatherSensor*, *homematicip.aio.device.AsyncDevice*

HMIP-SWO-B

class homematicip.aio.device.**AsyncWeatherSensorPlus**(*connection*)
Bases: *homematicip.device.WeatherSensorPlus*, *homematicip.aio.device.AsyncDevice*

HMIP-SWO-PL

class homematicip.aio.device.**AsyncWeatherSensorPro**(*connection*)
Bases: *homematicip.device.WeatherSensorPro*, *homematicip.aio.device.AsyncDevice*

HMIP-SWO-PR

class homematicip.aio.device.**AsyncWiredDimmer3**(*connection*)
Bases: *homematicip.device.WiredDimmer3*, *homematicip.aio.device.AsyncDimmer*

HMIPW-DRD3 (Homematic IP Wired Dimming Actuator – 3x channels)

class homematicip.aio.device.**AsyncWiredInput32**(*connection*)
Bases: *homematicip.device.WiredInput32*, *homematicip.aio.device.AsyncFullFlushContactInterface*

HMIPW-DRI32 (Homematic IP Wired Inbound module – 32x channels)

class homematicip.aio.device.**AsyncWiredSwitch8**(*connection*)
Bases: *homematicip.device.WiredSwitch8*, *homematicip.aio.device.AsyncSwitch*

HMIPW-DRS8 (Homematic IP Wired Switch Actuator – 8x channels)

homematicip.aio.group module

class homematicip.aio.group.**AsyncAlarmSwitchingGroup**(*connection*)
Bases: *homematicip.group.AlarmSwitchingGroup*, *homematicip.aio.group.AsyncGroup*

set_on_time(*onTimeSeconds*)

set_signal_acoustic(*signalAcoustic=<AcousticAlarmSignal.FREQUENCY_FALLING: 'FREQUENCY_FALLING'>*)

set_signal_optical(*signalOptical=<OpticalAlarmSignal.BLINKING_ALTERNATELY_REPEATING: 'BLINKING_ALTERNATELY_REPEATING'>*)

test_signal_acoustic(*signalAcoustic=<AcousticAlarmSignal.FREQUENCY_FALLING: 'FREQUENCY_FALLING'>*)

test_signal_optical(*signalOptical=<OpticalAlarmSignal.BLINKING_ALTERNATELY_REPEATING: 'BLINKING_ALTERNATELY_REPEATING'>*)

class homematicip.aio.group.**AsyncEnvironmentGroup**(*connection*)
Bases: *homematicip.group.EnvironmentGroup*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncExtendedLinkedShutterGroup**(*connection*)
Bases: *homematicip.group.ExtendedLinkedShutterGroup*, *homematicip.aio.group.AsyncGroup*

set_shutter_level(*level*)

set_shutter_stop()

set_slats_level(*slatsLevel=0.0*, *shutterLevel=None*)

class homematicip.aio.group.**AsyncExtendedLinkedSwitchingGroup**(*connection*)
Bases: *homematicip.group.ExtendedLinkedSwitchingGroup*, *homematicip.aio.group.AsyncSwitchGroupBase*

set_on_time(*onTimeSeconds*)

class homematicip.aio.group.**AsyncGroup**(*connection*)
Bases: *homematicip.group.Group*

delete()

set_label(*label*)

class homematicip.aio.group.**AsyncHeatingChangeoverGroup**(*connection*)
Bases: *homematicip.group.HeatingChangeoverGroup*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncHeatingCoolingDemandBoilerGroup**(*connection*)
Bases: *homematicip.group.HeatingCoolingDemandBoilerGroup*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncHeatingCoolingDemandGroup**(*connection*)
Bases: *homematicip.group.HeatingCoolingDemandGroup*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncHeatingCoolingDemandPumpGroup**(*connection*)
Bases: *homematicip.group.HeatingCoolingDemandPumpGroup*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncHeatingDehumidifierGroup**(*connection*)
Bases: *homematicip.group.HeatingDehumidifierGroup*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncHeatingExternalClockGroup**(*connection*)
Bases: *homematicip.group.HeatingExternalClockGroup*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncHeatingFailureAlertRuleGroup**(*connection*)
Bases: *homematicip.group.HeatingFailureAlertRuleGroup*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncHeatingGroup**(*connection*)
Bases: *homematicip.group.HeatingGroup*, *homematicip.aio.group.AsyncGroup*

set_active_profile(*index*)

set_boost(*enable=True*)

set_boost_duration(*duration: int*)

set_control_mode(*mode=<ClimateControlMode.AUTOMATIC: 'AUTOMATIC'>*)

set_point_temperature(*temperature*)

class homematicip.aio.group.**AsyncHeatingHumidyLimiterGroup**(*connection*)
Bases: *homematicip.group.HeatingHumidyLimiterGroup*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncHeatingTemperatureLimiterGroup**(*connection*)
Bases: *homematicip.group.HeatingTemperatureLimiterGroup*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncHotWaterGroup**(*connection*)
Bases: *homematicip.group.HotWaterGroup*, *homematicip.aio.group.AsyncGroup*

set_profile_mode(*profileMode: homematicip.base.enums.ProfileMode*)

class homematicip.aio.group.**AsyncHumidityWarningRuleGroup**(*connection*)
Bases: *homematicip.group.HumidityWarningRuleGroup*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncInboxGroup**(*connection*)
Bases: *homematicip.group.InboxGroup*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncLinkedSwitchingGroup**(*connection*)
Bases: *homematicip.group.LinkedSwitchingGroup*, *homematicip.aio.group.AsyncGroup*

set_light_group_switches(*devices*)

class homematicip.aio.group.**AsyncLockOutProtectionRule**(*connection*)
Bases: *homematicip.group.LockOutProtectionRule*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncMetaGroup**(*connection*)
Bases: *homematicip.group.MetaGroup*, *homematicip.aio.group.AsyncGroup*

a meta group is a "Room" inside the homematic configuration

class homematicip.aio.group.**AsyncOverHeatProtectionRule**(*connection*)
Bases: *homematicip.group.OverHeatProtectionRule*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncSecurityGroup**(*connection*)
Bases: *homematicip.group.SecurityGroup*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncSecurityZoneGroup**(*connection*)
Bases: *homematicip.group.SecurityZoneGroup*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncShutterProfile**(*connection*)
Bases: *homematicip.group.ShutterProfile*, *homematicip.aio.group.AsyncGroup*

set_profile_mode(*profileMode: homematicip.base.enums.ProfileMode*)

set_shutter_level(*level*)

set_shutter_stop()

set_slats_level(*slatsLevel*, *shutterlevel*)

class homematicip.aio.group.**AsyncShutterWindProtectionRule**(*connection*)
Bases: *homematicip.group.ShutterWindProtectionRule*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncSmokeAlarmDetectionRule**(*connection*)
Bases: *homematicip.group.SmokeAlarmDetectionRule*, *homematicip.aio.group.AsyncGroup*

class homematicip.aio.group.**AsyncSwitchGroupBase**(*connection*)
Bases: *homematicip.group.SwitchGroupBase*, *homematicip.aio.group.AsyncGroup*

set_switch_state(*on=True*)

turn_off()

turn_on()

class homematicip.aio.group.**AsyncSwitchingGroup**(*connection*)
Bases: *homematicip.group.SwitchingGroup*, *homematicip.aio.group.AsyncSwitchGroupBase*

set_shutter_level(*level*)

set_shutter_stop()

set_slats_level(*slatsLevel*, *shutterlevel*)

class homematicip.aio.group.**AsyncSwitchingProfileGroup**(*connection*)
Bases: *homematicip.group.SwitchingProfileGroup*, *homematicip.aio.group.AsyncGroup*

create(*label*)

set_group_channels()

set_profile_mode(*devices*, *automatic=True*)

homematicip.aio.home module

class homematicip.aio.home.**AsyncHome**(*loop*, *websession=None*)
Bases: *homematicip.home.Home*

this class represents the 'Async Home' of the homematic ip

activate_absence_permanent()
activates the absence forever

activate_absence_with_duration(*duration*)
activates the absence mode for a given time

Parameters **duration** (*int*) – the absence duration in minutes

activate_absence_with_period(*endtime*)
activates the absence mode until the given time

Parameters **endtime** (*datetime*) – the time when the absence should automatically be disabled

activate_vacation(*endtime*, *temperature*)
activates the vatation mode until the given time

Parameters

- **endtime** (*datetime*) – the time when the vatation mode should automatically be disabled
- **temperature** (*float*) – the settemperature during the vacation mode

deactivate_absence()
deactivates the absence mode immediately

deactivate_vacation()
deactivates the vacation mode immediately

delete_group(*group*)
deletes the given group from the cloud

Parameters **group** (Group) – the group to delete

disable_events()

download_configuration()
downloads the current configuration from the cloud

Returns the downloaded configuration or an errorCode

enable_events() → _asyncio.Task
Connects to the websocket. Returns a listening task.

get_OAuth_OTK()

get_current_state(*clearConfig: bool = False*)
downloads the current configuration and parses it into self

Parameters

- **clearConfig** (*bool*) – if set to true, this function will remove all old objects
- **self.devices, self.client, .. to have a fresh config instead of reparsing them** (*from*) –

get_security_journal()

init(*access_point_id*, *lookup=True*)

set_intrusion_alert_through_smoke_detectors(*activate=True*)
activate or deactivate if smoke detectors should "ring" during an alarm

Parameters **activate** (*bool*) – True will let the smoke detectors "ring" during an alarm

set_location(*city*, *latitude*, *longitude*)

set_pin(*newPin*, *oldPin=None*)
sets a new pin for the home

Parameters

- **newPin** (*str*) – the new pin
- **oldPin** (*str*) – optional, if there is currently a pin active it must be given here. Otherwise it will not be possible to set the new pin

Returns the result of the call

set_powermeter_unit_price(*price*)

set_security_zones_activation(*internal=True*, *external=True*)
this function will set the alarm system to armed or disable it

Parameters

- **internal** (*bool*) – activates/deactivates the internal zone
- **external** (*bool*) – activates/deactivates the external zone

Examples

arming while being at home

```
>>> home.set_security_zones_activation(False,True)
```

arming without being at home

```
>>> home.set_security_zones_activation(True,True)
```

disarming the alarm system

```
>>> home.set_security_zones_activation(False,False)
```

set_timezone(*timezone*)
sets the timezone for the AP. e.g. "Europe/Berlin" :param timezone: the new timezone :type timezone: str

set_zone_activation_delay(*delay*)

set_zones_device_assignment(*internal_devices*, *external_devices*)
sets the devices for the security zones :param internal_devices: the devices which should be used for the internal zone :type internal_devices: List[Device] :param external_devices: the devices which should be used for the external(hull) zone :type external_devices: List[Device]

Returns the result of _restCall

homematicip.aio.securityEvent module

class homematicip.aio.securityEvent.**AsyncAccessPointConnectedEvent**(*connection*)
Bases: *homematicip.securityEvent.AccessPointConnectedEvent*, *homematicip.aio.securityEvent.AsyncSecurityEvent*

class homematicip.aio.securityEvent.**AsyncAccessPointDisconnectedEvent**(*connection*)
Bases: *homematicip.securityEvent.AccessPointDisconnectedEvent*, *homematicip.aio.securityEvent.AsyncSecurityEvent*

class homematicip.aio.securityEvent.**AsyncActivationChangedEvent**(*connection*)
Bases: *homematicip.securityEvent.ActivationChangedEvent*, *homematicip.aio.securityEvent.AsyncSecurityZoneEvent*

class homematicip.aio.securityEvent.**AsyncExternalTriggeredEvent**(*connection*)
Bases: *homematicip.securityEvent.ExternalTriggeredEvent*, *homematicip.aio.securityEvent.AsyncSecurityEvent*

class homematicip.aio.securityEvent.**AsyncMainsFailureEvent**(*connection*)
Bases: *homematicip.securityEvent.MainsFailureEvent*, *homematicip.aio.securityEvent.AsyncSecurityEvent*

class homematicip.aio.securityEvent.**AsyncMoistureDetectionEvent**(*connection*)
Bases: *homematicip.securityEvent.MoistureDetectionEvent*, *homematicip.aio.securityEvent.AsyncSecurityEvent*

class homematicip.aio.securityEvent.**AsyncOfflineAlarmEvent**(*connection*)
Bases: *homematicip.securityEvent.OfflineAlarmEvent*, *homematicip.aio.securityEvent.AsyncSecurityEvent*

class homematicip.aio.securityEvent.**AsyncOfflineWaterDetectionEvent**(*connection*)
Bases: *homematicip.securityEvent.OfflineWaterDetectionEvent*, *homematicip.aio.securityEvent.AsyncSecurityEvent*

class homematicip.aio.securityEvent.**AsyncSabotageEvent**(*connection*)
Bases: *homematicip.securityEvent.SabotageEvent*, *homematicip.aio.securityEvent.AsyncSecurityEvent*

class homematicip.aio.securityEvent.**AsyncSecurityEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityEvent*

this class represents a security event

class homematicip.aio.securityEvent.**AsyncSecurityZoneEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityZoneEvent*, *homematicip.aio.securityEvent.AsyncSecurityEvent*

This class will be used by other events which are just adding “securityZoneValues”

class homematicip.aio.securityEvent.**AsyncSensorEvent**(*connection*)
Bases: *homematicip.securityEvent.SensorEvent*, *homematicip.aio.securityEvent.AsyncSecurityEvent*

class homematicip.aio.securityEvent.**AsyncSilenceChangedEvent**(*connection*)
Bases: *homematicip.securityEvent.SilenceChangedEvent*, *homematicip.aio.securityEvent.AsyncSecurityZoneEvent*

class homematicip.aio.securityEvent.**AsyncSmokeAlarmEvent**(*connection*)
Bases: *homematicip.securityEvent.SmokeAlarmEvent*, *homematicip.aio.securityEvent.AsyncSecurityEvent*

class homematicip.aio.securityEvent.**AsyncWaterDetectionEvent**(*connection*)
Bases: *homematicip.securityEvent.WaterDetectionEvent*, *homematicip.aio.securityEvent.AsyncSecurityEvent*

Module contents

homematicip.base package

Submodules

homematicip.base.base_connection module

class homematicip.base.base_connection.**BaseConnection**
Bases: object

Base connection class.

Threaded and Async connection class must inherit from this.

auth_token

clientCharacteristics

clientauth_token

init(*accesspoint_id*, *lookup=True*, ***kwargs*)

set_auth_token(*auth_token*)

set_token_and_characteristics(*accesspoint_id*)

urlREST

urlWebSocket

exception homematicip.base.base_connection.**HmipConnectionError**
Bases: Exception

exception homematicip.base.base_connection.**HmipServerCloseError**
Bases: *homematicip.base.base_connection.HmipConnectionError*

exception homematicip.base.base_connection.**HmipWrongHttpStatusError**(*status_code=None*)
Bases: *homematicip.base.base_connection.HmipConnectionError*

homematicip.base.constants module

homematicip.base.enums module

class homematicip.base.enums.**AbsenceType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

NOT_ABSENT = 'NOT_ABSENT'

PARTY = 'PARTY'

PERIOD = 'PERIOD'

PERMANENT = 'PERMANENT'

VACATION = 'VACATION'

class homematicip.base.enums.**AccelerationSensorMode**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

ANY_MOTION = 'ANY_MOTION'

FLAT_DECT = 'FLAT_DECT'

class homematicip.base.enums.**AccelerationSensorNeutralPosition**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

HORIZONTAL = 'HORIZONTAL'

VERTICAL = 'VERTICAL'

class homematicip.base.enums.**AccelerationSensorSensitivity**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

SENSOR_RANGE_16G = 'SENSOR_RANGE_16G'

SENSOR_RANGE_2G = 'SENSOR_RANGE_2G'

SENSOR_RANGE_2G_2PLUS_SENSE = 'SENSOR_RANGE_2G_2PLUS_SENSE'

SENSOR_RANGE_2G_PLUS_SENS = 'SENSOR_RANGE_2G_PLUS_SENS'

SENSOR_RANGE_4G = 'SENSOR_RANGE_4G'

SENSOR_RANGE_8G = 'SENSOR_RANGE_8G'

class homematicip.base.enums.**AcousticAlarmSignal**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

DELAYED_EXTERNALLY_ARMED = 'DELAYED_EXTERNALLY_ARMED'

DELAYED_INTERNALLY_ARMED = 'DELAYED_INTERNALLY_ARMED'

DISABLE_ACOUSTIC_SIGNAL = 'DISABLE_ACOUSTIC_SIGNAL'

DISARMED = 'DISARMED'

ERROR = 'ERROR'

EVENT = 'EVENT'

EXTERNALLY_ARMED = 'EXTERNALLY_ARMED'

FREQUENCY_ALTERNATING_LOW_HIGH = 'FREQUENCY_ALTERNATING_LOW_HIGH'

FREQUENCY_ALTERNATING_LOW_MID_HIGH = 'FREQUENCY_ALTERNATING_LOW_MID_HIGH'

FREQUENCY_FALLING = 'FREQUENCY_FALLING'

FREQUENCY_HIGHON_LONGOFF = 'FREQUENCY_HIGHON_LONGOFF'

FREQUENCY_HIGHON_OFF = 'FREQUENCY_HIGHON_OFF'

FREQUENCY_LOWON_LONGOFF_HIGHON_LONGOFF = 'FREQUENCY_LOWON_LONGOFF_HIGHON_LONGOFF'

FREQUENCY_LOWON_OFF_HIGHON_OFF = 'FREQUENCY_LOWON_OFF_HIGHON_OFF'

FREQUENCY_RISING = 'FREQUENCY_RISING'

FREQUENCY_RISING_AND_FALLING = 'FREQUENCY_RISING_AND_FALLING'

INTERNALLY_ARMED = 'INTERNALLY_ARMED'

LOW_BATTERY = 'LOW_BATTERY'

class homematicip.base.enums.**AcousticAlarmTiming**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

ONCE_PER_MINUTE = 'ONCE_PER_MINUTE'

PERMANENT = 'PERMANENT'

SIX_MINUTES = 'SIX_MINUTES'

THREE_MINUTES = 'THREE_MINUTES'

class homematicip.base.enums.**AlarmContactType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

PASSIVE_GLASS_BREAKAGE_DETECTOR = 'PASSIVE_GLASS_BREAKAGE_DETECTOR'

WINDOW_DOOR_CONTACT = 'WINDOW_DOOR_CONTACT'

class homematicip.base.enums.**AlarmSignalType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

FULL_ALARM = 'FULL_ALARM'

NO_ALARM = 'NO_ALARM'

SILENT_ALARM = 'SILENT_ALARM'

class homematicip.base.enums.**ApExchangeState**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

DONE = 'DONE'

IN_PROGRESS = 'IN_PROGRESS'

NONE = 'NONE'

REJECTED = 'REJECTED'

REQUESTED = 'REQUESTED'

class homematicip.base.enums.**AutoNameEnum**(**args*, ***kwds*)
Bases: `str`, `aenum.Enum`

auto() will generate the name of the attribute as value

from_str = <bound method AutoNameEnum.from_str of <aenum 'AutoNameEnum'>>

class homematicip.base.enums.**AutomationRuleType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

SIMPLE = 'SIMPLE'

class homematicip.base.enums.**BinaryBehaviorType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

NORMALLY_CLOSE = 'NORMALLY_CLOSE'

NORMALLY_OPEN = 'NORMALLY_OPEN'

class homematicip.base.enums.**ClientType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

APP = 'APP'

C2C = 'C2C'

class homematicip.base.enums.**ClimateControlDisplay**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

ACTUAL = 'ACTUAL'

ACTUAL_HUMIDITY = 'ACTUAL_HUMIDITY'

SETPOINT = 'SETPOINT'

class homematicip.base.enums.**ClimateControlMode**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

AUTOMATIC = 'AUTOMATIC'

ECO = 'ECO'

MANUAL = 'MANUAL'

class homematicip.base.enums.**ConnectionType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

HMIP_LAN = 'HMIP_LAN'

HMIP_RF = 'HMIP_RF'

HMIP_WIRED = 'HMIP_WIRED'

HMIP_WLAN = 'HMIP_WLAN'

class homematicip.base.enums.**ContactType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

NORMALLY_CLOSE = 'NORMALLY_CLOSE'

NORMALLY_OPEN = 'NORMALLY_OPEN'

class homematicip.base.enums.**DeviceType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

ACCELERATION_SENSOR = 'ACCELERATION_SENSOR'

ALARM_SIREN_INDOOR = 'ALARM_SIREN_INDOOR'

ALARM_SIREN_OUTDOOR = 'ALARM_SIREN_OUTDOOR'

BLIND_MODULE = 'BLIND_MODULE'

BRAND_BLIND = 'BRAND_BLIND'

```
BRAND_DIMMER = 'BRAND_DIMMER'
BRAND_PUSH_BUTTON = 'BRAND_PUSH_BUTTON'
BRAND_SHUTTER = 'BRAND_SHUTTER'
BRAND_SWITCH_MEASURING = 'BRAND_SWITCH_MEASURING'
BRAND_SWITCH_NOTIFICATION_LIGHT = 'BRAND_SWITCH_NOTIFICATION_LIGHT'
BRAND_WALL_MOUNTED_THERMOSTAT = 'BRAND_WALL_MOUNTED_THERMOSTAT'
DEVICE = 'DEVICE'
DIN_RAIL_BLIND_4 = 'DIN_RAIL_BLIND_4'
DIN_RAIL_DIMMER_3 = 'DIN_RAIL_DIMMER_3'
DIN_RAIL_SWITCH = 'DIN_RAIL_SWITCH'
DIN_RAIL_SWITCH_4 = 'DIN_RAIL_SWITCH_4'
FLOOR_TERMINAL_BLOCK_10 = 'FLOOR_TERMINAL_BLOCK_10'
FLOOR_TERMINAL_BLOCK_12 = 'FLOOR_TERMINAL_BLOCK_12'
FLOOR_TERMINAL_BLOCK_6 = 'FLOOR_TERMINAL_BLOCK_6'
FULL_FLUSH_BLIND = 'FULL_FLUSH_BLIND'
FULL_FLUSH_CONTACT_INTERFACE = 'FULL_FLUSH_CONTACT_INTERFACE'
FULL_FLUSH_CONTACT_INTERFACE_6 = 'FULL_FLUSH_CONTACT_INTERFACE_6'
FULL_FLUSH_DIMMER = 'FULL_FLUSH_DIMMER'
FULL_FLUSH_INPUT_SWITCH = 'FULL_FLUSH_INPUT_SWITCH'
FULL_FLUSH_SHUTTER = 'FULL_FLUSH_SHUTTER'
FULL_FLUSH_SWITCH_MEASURING = 'FULL_FLUSH_SWITCH_MEASURING'
HEATING_SWITCH_2 = 'HEATING_SWITCH_2'
HEATING_THERMOSTAT = 'HEATING_THERMOSTAT'
HEATING_THERMOSTAT_COMPACT = 'HEATING_THERMOSTAT_COMPACT'
HEATING_THERMOSTAT_EVO = 'HEATING_THERMOSTAT_EVO'
HOERMANN_DRIVES_MODULE = 'HOERMANN_DRIVES_MODULE'
HOME_CONTROL_ACCESS_POINT = 'HOME_CONTROL_ACCESS_POINT'
KEY_REMOTE_CONTROL_4 = 'KEY_REMOTE_CONTROL_4'
KEY_REMOTE_CONTROL_ALARM = 'KEY_REMOTE_CONTROL_ALARM'
LIGHT_SENSOR = 'LIGHT_SENSOR'
MOTION_DETECTOR_INDOOR = 'MOTION_DETECTOR_INDOOR'
MOTION_DETECTOR_OUTDOOR = 'MOTION_DETECTOR_OUTDOOR'
MOTION_DETECTOR_PUSH_BUTTON = 'MOTION_DETECTOR_PUSH_BUTTON'
MULTI_IO_BOX = 'MULTI_IO_BOX'
OPEN_COLLECTOR_8_MODULE = 'OPEN_COLLECTOR_8_MODULE'
PASSAGE_DETECTOR = 'PASSAGE_DETECTOR'
```

```
PLUGABLE_SWITCH = 'PLUGABLE_SWITCH'
PLUGABLE_SWITCH_MEASURING = 'PLUGABLE_SWITCH_MEASURING'
PLUGGABLE_DIMMER = 'PLUGGABLE_DIMMER'
PLUGGABLE_MAINS_FAILURE_SURVEILLANCE = 'PLUGGABLE_MAINS_FAILURE_SURVEILLANCE'
PRESENCE_DETECTOR_INDOOR = 'PRESENCE_DETECTOR_INDOOR'
PRINTED_CIRCUIT_BOARD_SWITCH_2 = 'PRINTED_CIRCUIT_BOARD_SWITCH_2'
PRINTED_CIRCUIT_BOARD_SWITCH_BATTERY = 'PRINTED_CIRCUIT_BOARD_SWITCH_BATTERY'
PUSH_BUTTON = 'PUSH_BUTTON'
PUSH_BUTTON_6 = 'PUSH_BUTTON_6'
PUSH_BUTTON_FLAT = 'PUSH_BUTTON_FLAT'
RAIN_SENSOR = 'RAIN_SENSOR'
REMOTE_CONTROL_8 = 'REMOTE_CONTROL_8'
REMOTE_CONTROL_8_MODULE = 'REMOTE_CONTROL_8_MODULE'
ROOM_CONTROL_DEVICE = 'ROOM_CONTROL_DEVICE'
ROOM_CONTROL_DEVICE_ANALOG = 'ROOM_CONTROL_DEVICE_ANALOG'
ROTARY_HANDLE_SENSOR = 'ROTARY_HANDLE_SENSOR'
SHUTTER_CONTACT = 'SHUTTER_CONTACT'
SHUTTER_CONTACT_INTERFACE = 'SHUTTER_CONTACT_INTERFACE'
SHUTTER_CONTACT_INVISIBLE = 'SHUTTER_CONTACT_INVISIBLE'
SHUTTER_CONTACT_MAGNETIC = 'SHUTTER_CONTACT_MAGNETIC'
SHUTTER_CONTACT_OPTICAL_PLUS = 'SHUTTER_CONTACT_OPTICAL_PLUS'
SMOKE_DETECTOR = 'SMOKE_DETECTOR'
TEMPERATURE_HUMIDITY_SENSOR = 'TEMPERATURE_HUMIDITY_SENSOR'
TEMPERATURE_HUMIDITY_SENSOR_DISPLAY = 'TEMPERATURE_HUMIDITY_SENSOR_DISPLAY'
TEMPERATURE_HUMIDITY_SENSOR_OUTDOOR = 'TEMPERATURE_HUMIDITY_SENSOR_OUTDOOR'
TEMPERATURE_SENSOR_2_EXTERNAL_DELTA = 'TEMPERATURE_SENSOR_2_EXTERNAL_DELTA'
TILT_VIBRATION_SENSOR = 'TILT_VIBRATION_SENSOR'
TORMATIC_MODULE = 'TORMATIC_MODULE'
WALL_MOUNTED_THERMOSTAT_BASIC_HUMIDITY = 'WALL_MOUNTED_THERMOSTAT_BASIC_HUMIDITY'
WALL_MOUNTED_THERMOSTAT_PRO = 'WALL_MOUNTED_THERMOSTAT_PRO'
WATER_SENSOR = 'WATER_SENSOR'
WEATHER_SENSOR = 'WEATHER_SENSOR'
WEATHER_SENSOR_PLUS = 'WEATHER_SENSOR_PLUS'
WEATHER_SENSOR_PRO = 'WEATHER_SENSOR_PRO'
WIRED_DIMMER_3 = 'WIRED_DIMMER_3'
WIRED_INPUT_32 = 'WIRED_INPUT_32'
```

WIRED_SWITCH_8 = 'WIRED_SWITCH_8'

class homematicip.base.enums.**DeviceUpdateState**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

BACKGROUND_UPDATE_NOT_SUPPORTED = 'BACKGROUND_UPDATE_NOT_SUPPORTED'

TRANSFERING_UPDATE = 'TRANSFERING_UPDATE'

UPDATE_AUTHORIZED = 'UPDATE_AUTHORIZED'

UPDATE_AVAILABLE = 'UPDATE_AVAILABLE'

UP_TO_DATE = 'UP_TO_DATE'

class homematicip.base.enums.**DeviceUpdateStrategy**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

AUTOMATICALLY_IF_POSSIBLE = 'AUTOMATICALLY_IF_POSSIBLE'

MANUALLY = 'MANUALLY'

class homematicip.base.enums.**DoorCommand**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

CLOSE = 'CLOSE'

OPEN = 'OPEN'

PARTIAL_OPEN = 'PARTIAL_OPEN'

STOP = 'STOP'

class homematicip.base.enums.**DoorState**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

CLOSED = 'CLOSED'

OPEN = 'OPEN'

POSITION_UNKNOWN = 'POSITION_UNKNOWN'

VENTILATION_POSITION = 'VENTILATION_POSITION'

class homematicip.base.enums.**DriveSpeed**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

CREEP_SPEED = 'CREEP_SPEED'

NOMINAL_SPEED = 'NOMINAL_SPEED'

OPTIONAL_SPEED = 'OPTIONAL_SPEED'

SLOW_SPEED = 'SLOW_SPEED'

class homematicip.base.enums.**EcoDuration**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

FOUR = 'FOUR'

ONE = 'ONE'

PERMANENT = 'PERMANENT'

SIX = 'SIX'

TWO = 'TWO'

class homematicip.base.enums.**EventType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

CLIENT_ADDED = 'CLIENT_ADDED'

CLIENT_CHANGED = 'CLIENT_CHANGED'

CLIENT_REMOVED = 'CLIENT_REMOVED'

DEVICE_ADDED = 'DEVICE_ADDED'

DEVICE_CHANGED = 'DEVICE_CHANGED'

DEVICE_REMOVED = 'DEVICE_REMOVED'

GROUP_ADDED = 'GROUP_ADDED'

GROUP_CHANGED = 'GROUP_CHANGED'

GROUP_REMOVED = 'GROUP_REMOVED'

HOME_CHANGED = 'HOME_CHANGED'

SECURITY_JOURNAL_CHANGED = 'SECURITY_JOURNAL_CHANGED'

class homematicip.base.enums.**FunctionalChannelType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

ACCELERATION_SENSOR_CHANNEL = 'ACCELERATION_SENSOR_CHANNEL'

ACCESS_CONTROLLER_CHANNEL = 'ACCESS_CONTROLLER_CHANNEL'

ALARM_SIREN_CHANNEL = 'ALARM_SIREN_CHANNEL'

ANALOG_OUTPUT_CHANNEL = 'ANALOG_OUTPUT_CHANNEL'

ANALOG_ROOM_CONTROL_CHANNEL = 'ANALOG_ROOM_CONTROL_CHANNEL'

BLIND_CHANNEL = 'BLIND_CHANNEL'

CHANGE_OVER_CHANNEL = 'CHANGE_OVER_CHANNEL'

CLIMATE_SENSOR_CHANNEL = 'CLIMATE_SENSOR_CHANNEL'

CONTACT_INTERFACE_CHANNEL = 'CONTACT_INTERFACE_CHANNEL'

DEHUMIDIFIER_DEMAND_CHANNEL = 'DEHUMIDIFIER_DEMAND_CHANNEL'

DEVICE_BASE = 'DEVICE_BASE'

DEVICE_BASE_FLOOR_HEATING = 'DEVICE_BASE_FLOOR_HEATING'

DEVICE_GLOBAL_PUMP_CONTROL = 'DEVICE_GLOBAL_PUMP_CONTROL'

DEVICE_INCORRECT_POSITIONED = 'DEVICE_INCORRECT_POSITIONED'

DEVICE_OPERATIONLOCK = 'DEVICE_OPERATIONLOCK'

```
DEVICE_PERMANENT_FULL_RX = 'DEVICE_PERMANENT_FULL_RX'
DEVICE_RECHARGEABLE_WITH_SABOTAGE = 'DEVICE_RECHARGEABLE_WITH_SABOTAGE'
DEVICE_SABOTAGE = 'DEVICE_SABOTAGE'
DIMMER_CHANNEL = 'DIMMER_CHANNEL'
DOOR_CHANNEL = 'DOOR_CHANNEL'
FLOOR_TERMINAL_BLOCK_CHANNEL = 'FLOOR_TERMINAL_BLOCK_CHANNEL'
FLOOR_TERMINAL_BLOCK_LOCAL_PUMP_CHANNEL = 'FLOOR_TERMINAL_BLOCK_LOCAL_PUMP_CHANNEL'
FLOOR_TERMINAL_BLOCK_MECHANIC_CHANNEL = 'FLOOR_TERMINAL_BLOCK_MECHANIC_CHANNEL'
FUNCTIONAL_CHANNEL = 'FUNCTIONAL_CHANNEL'
GENERIC_INPUT_CHANNEL = 'GENERIC_INPUT_CHANNEL'
HEATING_THERMOSTAT_CHANNEL = 'HEATING_THERMOSTAT_CHANNEL'
HEAT_DEMAND_CHANNEL = 'HEAT_DEMAND_CHANNEL'
INTERNAL_SWITCH_CHANNEL = 'INTERNAL_SWITCH_CHANNEL'
LIGHT_SENSOR_CHANNEL = 'LIGHT_SENSOR_CHANNEL'
MAINS_FAILURE_CHANNEL = 'MAINS_FAILURE_CHANNEL'
MOTION_DETECTION_CHANNEL = 'MOTION_DETECTION_CHANNEL'
MULTI_MODE_INPUT_BLIND_CHANNEL = 'MULTI_MODE_INPUT_BLIND_CHANNEL'
MULTI_MODE_INPUT_CHANNEL = 'MULTI_MODE_INPUT_CHANNEL'
MULTI_MODE_INPUT_DIMMER_CHANNEL = 'MULTI_MODE_INPUT_DIMMER_CHANNEL'
MULTI_MODE_INPUT_SWITCH_CHANNEL = 'MULTI_MODE_INPUT_SWITCH_CHANNEL'
NOTIFICATION_LIGHT_CHANNEL = 'NOTIFICATION_LIGHT_CHANNEL'
PASSAGE_DETECTOR_CHANNEL = 'PASSAGE_DETECTOR_CHANNEL'
PRESENCE_DETECTION_CHANNEL = 'PRESENCE_DETECTION_CHANNEL'
RAIN_DETECTION_CHANNEL = 'RAIN_DETECTION_CHANNEL'
ROTARY_HANDLE_CHANNEL = 'ROTARY_HANDLE_CHANNEL'
SHADING_CHANNEL = 'SHADING_CHANNEL'
SHUTTER_CHANNEL = 'SHUTTER_CHANNEL'
SHUTTER_CONTACT_CHANNEL = 'SHUTTER_CONTACT_CHANNEL'
SINGLE_KEY_CHANNEL = 'SINGLE_KEY_CHANNEL'
SMOKE_DETECTOR_CHANNEL = 'SMOKE_DETECTOR_CHANNEL'
SWITCH_CHANNEL = 'SWITCH_CHANNEL'
SWITCH_MEASURING_CHANNEL = 'SWITCH_MEASURING_CHANNEL'
TEMPERATURE_SENSOR_2_EXTERNAL_DELTA_CHANNEL = 'TEMPERATURE_SENSOR_2_EXTERNAL_DELTA_CHA
TILT_VIBRATION_SENSOR_CHANNEL = 'TILT_VIBRATION_SENSOR_CHANNEL'
WALL_MOUNTED_THERMOSTAT_PRO_CHANNEL = 'WALL_MOUNTED_THERMOSTAT_PRO_CHANNEL'
WALL_MOUNTED_THERMOSTAT_WITHOUT_DISPLAY_CHANNEL = 'WALL_MOUNTED_THERMOSTAT_WITHOUT_DIS
```

WATER_SENSOR_CHANNEL = 'WATER_SENSOR_CHANNEL'

WEATHER_SENSOR_CHANNEL = 'WEATHER_SENSOR_CHANNEL'

WEATHER_SENSOR_PLUS_CHANNEL = 'WEATHER_SENSOR_PLUS_CHANNEL'

WEATHER_SENSOR_PRO_CHANNEL = 'WEATHER_SENSOR_PRO_CHANNEL'

class homematicip.base.enums.**FunctionalHomeType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

ACCESS_CONTROL = 'ACCESS_CONTROL'

INDOOR_CLIMATE = 'INDOOR_CLIMATE'

LIGHT_AND_SHADOW = 'LIGHT_AND_SHADOW'

SECURITY_AND_ALARM = 'SECURITY_AND_ALARM'

WEATHER_AND_ENVIRONMENT = 'WEATHER_AND_ENVIRONMENT'

class homematicip.base.enums.**GroupType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

ALARM_SWITCHING = 'ALARM_SWITCHING'

ENVIRONMENT = 'ENVIRONMENT'

EXTENDED_LINKED_SHUTTER = 'EXTENDED_LINKED_SHUTTER'

EXTENDED_LINKED_SWITCHING = 'EXTENDED_LINKED_SWITCHING'

GROUP = 'GROUP'

HEATING = 'HEATING'

HEATING_CHANGEOVER = 'HEATING_CHANGEOVER'

HEATING_COOLING_DEMAND = 'HEATING_COOLING_DEMAND'

HEATING_COOLING_DEMAND_BOILER = 'HEATING_COOLING_DEMAND_BOILER'

HEATING_COOLING_DEMAND_PUMP = 'HEATING_COOLING_DEMAND_PUMP'

HEATING_DEHUMIDIFIER = 'HEATING_DEHUMIDIFIER'

HEATING_EXTERNAL_CLOCK = 'HEATING_EXTERNAL_CLOCK'

HEATING_FAILURE_ALERT_RULE_GROUP = 'HEATING_FAILURE_ALERT_RULE_GROUP'

HEATING_HUMIDITY_LIMITER = 'HEATING_HUMIDITY_LIMITER'

HEATING_TEMPERATURE_LIMITER = 'HEATING_TEMPERATURE_LIMITER'

HOT_WATER = 'HOT_WATER'

HUMIDITY_WARNING_RULE_GROUP = 'HUMIDITY_WARNING_RULE_GROUP'

INBOX = 'INBOX'

LINKED_SWITCHING = 'LINKED_SWITCHING'

LOCK_OUT_PROTECTION_RULE = 'LOCK_OUT_PROTECTION_RULE'

OVER_HEAT_PROTECTION_RULE = 'OVER_HEAT_PROTECTION_RULE'

SECURITY = 'SECURITY'

SECURITY_BACKUP_ALARM_SWITCHING = 'SECURITY_BACKUP_ALARM_SWITCHING'

SECURITY_ZONE = 'SECURITY_ZONE'

SHUTTER_PROFILE = 'SHUTTER_PROFILE'

SHUTTER_WIND_PROTECTION_RULE = 'SHUTTER_WIND_PROTECTION_RULE'

SMOKE_ALARM_DETECTION_RULE = 'SMOKE_ALARM_DETECTION_RULE'

SWITCHING = 'SWITCHING'

SWITCHING_PROFILE = 'SWITCHING_PROFILE'

class homematicip.base.enums.**GroupVisibility**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

INVISIBLE_CONTROL = 'INVISIBLE_CONTROL'

INVISIBLE_GROUP_AND_CONTROL = 'INVISIBLE_GROUP_AND_CONTROL'

VISIBLE = 'VISIBLE'

class homematicip.base.enums.**HeatingFailureValidationType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

HEATING_FAILURE_ALARM = 'HEATING_FAILURE_ALARM'

HEATING_FAILURE_WARNING = 'HEATING_FAILURE_WARNING'

NO_HEATING_FAILURE = 'NO_HEATING_FAILURE'

class homematicip.base.enums.**HeatingLoadType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

LOAD_BALANCING = 'LOAD_BALANCING'

LOAD_COLLECTION = 'LOAD_COLLECTION'

class homematicip.base.enums.**HeatingValveType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

NORMALLY_CLOSE = 'NORMALLY_CLOSE'

NORMALLY_OPEN = 'NORMALLY_OPEN'

class homematicip.base.enums.**HomeUpdateState**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

PERFORMING_UPDATE = 'PERFORMING_UPDATE'

PERFORM_UPDATE_SENT = 'PERFORM_UPDATE_SENT'

UPDATE_AVAILABLE = 'UPDATE_AVAILABLE'

UP_TO_DATE = 'UP_TO_DATE'

class homematicip.base.enums.**HumidityValidationType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

GREATER_LOWER_LESSER_UPPER_THRESHOLD = 'GREATER_LOWER_LESSER_UPPER_THRESHOLD'

GREATER_UPPER_THRESHOLD = 'GREATER_UPPER_THRESHOLD'

LESSER_LOWER_THRESHOLD = 'LESSER_LOWER_THRESHOLD'

class homematicip.base.enums.**LiveUpdateState**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

LIVE_UPDATE_NOT_SUPPORTED = 'LIVE_UPDATE_NOT_SUPPORTED'

UPDATE_AVAILABLE = 'UPDATE_AVAILABLE'

UPDATE_INCOMPLETE = 'UPDATE_INCOMPLETE'

UP_TO_DATE = 'UP_TO_DATE'

class homematicip.base.enums.**MotionDetectionSendInterval**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

SECONDS_120 = 'SECONDS_120'

SECONDS_240 = 'SECONDS_240'

SECONDS_30 = 'SECONDS_30'

SECONDS_480 = 'SECONDS_480'

SECONDS_60 = 'SECONDS_60'

class homematicip.base.enums.**MultiModeInputMode**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

BINARY_BEHAVIOR = 'BINARY_BEHAVIOR'

KEY_BEHAVIOR = 'KEY_BEHAVIOR'

SWITCH_BEHAVIOR = 'SWITCH_BEHAVIOR'

class homematicip.base.enums.**NotificationSoundType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

SOUND_LONG = 'SOUND_LONG'

SOUND_NO_SOUND = 'SOUND_NO_SOUND'

SOUND_SHORT = 'SOUND_SHORT'

SOUND_SHORT_SHORT = 'SOUND_SHORT_SHORT'

class homematicip.base.enums.**OpticalAlarmSignal**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

BLINKING_ALTERNATELY_REPEATING = 'BLINKING_ALTERNATELY_REPEATING'

BLINKING_BOTH_REPEATING = 'BLINKING_BOTH_REPEATING'

CONFIRMATION_SIGNAL_0 = 'CONFIRMATION_SIGNAL_0'

CONFIRMATION_SIGNAL_1 = 'CONFIRMATION_SIGNAL_1'

CONFIRMATION_SIGNAL_2 = 'CONFIRMATION_SIGNAL_2'

DISABLE_OPTICAL_SIGNAL = 'DISABLE_OPTICAL_SIGNAL'

DOUBLE_FLASHING_REPEATING = 'DOUBLE_FLASHING_REPEATING'

FLASHING_BOTH_REPEATING = 'FLASHING_BOTH_REPEATING'

class homematicip.base.enums.**PassageDirection**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

LEFT = 'LEFT'

RIGHT = 'RIGHT'

class homematicip.base.enums.**ProfileMode**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

AUTOMATIC = 'AUTOMATIC'

MANUAL = 'MANUAL'

class homematicip.base.enums.**RGBColorState**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

BLACK = 'BLACK'

BLUE = 'BLUE'

GREEN = 'GREEN'

PURPLE = 'PURPLE'

RED = 'RED'

TURQUOISE = 'TURQUOISE'

WHITE = 'WHITE'

YELLOW = 'YELLOW'

class homematicip.base.enums.**SecurityEventType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

ACCESS_POINT_CONNECTED = 'ACCESS_POINT_CONNECTED'

ACCESS_POINT_DISCONNECTED = 'ACCESS_POINT_DISCONNECTED'

ACTIVATION_CHANGED = 'ACTIVATION_CHANGED'

EXTERNAL_TRIGGERED = 'EXTERNAL_TRIGGERED'

MAINS_FAILURE_EVENT = 'MAINS_FAILURE_EVENT'

MOISTURE_DETECTION_EVENT = 'MOISTURE_DETECTION_EVENT'

OFFLINE_ALARM = 'OFFLINE_ALARM'

OFFLINE_WATER_DETECTION_EVENT = 'OFFLINE_WATER_DETECTION_EVENT'

SABOTAGE = 'SABOTAGE'

SENSOR_EVENT = 'SENSOR_EVENT'

SILENCE_CHANGED = 'SILENCE_CHANGED'

SMOKE_ALARM = 'SMOKE_ALARM'

WATER_DETECTION_EVENT = 'WATER_DETECTION_EVENT'

class homematicip.base.enums.**SecurityZoneActivationMode**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

ACTIVATION_IF_ALL_IN_VALID_STATE = 'ACTIVATION_IF_ALL_IN_VALID_STATE'

ACTIVATION_WITH_DEVICE_IGNORELIST = 'ACTIVATION_WITH_DEVICE_IGNORELIST'

class homematicip.base.enums.**ShadingPackagePosition**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

BOTTOM = 'BOTTOM'

CENTER = 'CENTER'

LEFT = 'LEFT'

NOT_USED = 'NOT_USED'

RIGHT = 'RIGHT'

SPLIT = 'SPLIT'

TDBU = 'TDBU'

TOP = 'TOP'

class homematicip.base.enums.**ShadingStateType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

MIXED = 'MIXED'

NOT_EXISTENT = 'NOT_EXISTENT'

NOT_POSSIBLE = 'NOT_POSSIBLE'

NOT_USED = 'NOT_USED'

POSITION_USED = 'POSITION_USED'

TILT_USED = 'TILT_USED'

class homematicip.base.enums.**SmokeDetectorAlarmType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

IDLE_OFF = 'IDLE_OFF'

INTRUSION_ALARM = 'INTRUSION_ALARM'

PRIMARY_ALARM = 'PRIMARY_ALARM'

SECONDARY_ALARM = 'SECONDARY_ALARM'

class homematicip.base.enums.**ValveState**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

ADAPTION_DONE = 'ADAPTION_DONE'

ADAPTION_IN_PROGRESS = 'ADAPTION_IN_PROGRESS'

ADJUSTMENT_TOO_BIG = 'ADJUSTMENT_TOO_BIG'

ADJUSTMENT_TOO_SMALL = 'ADJUSTMENT_TOO_SMALL'

ERROR_POSITION = 'ERROR_POSITION'

RUN_TO_START = 'RUN_TO_START'

STATE_NOT_AVAILABLE = 'STATE_NOT_AVAILABLE'

TOO_TIGHT = 'TOO_TIGHT'

WAIT_FOR_ADAPTION = 'WAIT_FOR_ADAPTION'

class homematicip.base.enums.**WaterAlarmTrigger**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

MOISTURE_DETECTION = 'MOISTURE_DETECTION'

NO_ALARM = 'NO_ALARM'

WATER_DETECTION = 'WATER_DETECTION'

WATER_MOISTURE_DETECTION = 'WATER_MOISTURE_DETECTION'

class homematicip.base.enums.**WeatherCondition**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

CLEAR = 'CLEAR'

CLOUDY = 'CLOUDY'

CLOUDY_WITH_RAIN = 'CLOUDY_WITH_RAIN'

CLOUDY_WITH_SNOW_RAIN = 'CLOUDY_WITH_SNOW_RAIN'

FOGGY = 'FOGGY'

HEAVILY_CLOUDY = 'HEAVILY_CLOUDY'

HEAVILY_CLOUDY_WITH_RAIN = 'HEAVILY_CLOUDY_WITH_RAIN'

HEAVILY_CLOUDY_WITH_RAIN_AND_THUNDER = 'HEAVILY_CLOUDY_WITH_RAIN_AND_THUNDER'

HEAVILY_CLOUDY_WITH_SNOW = 'HEAVILY_CLOUDY_WITH_SNOW'

HEAVILY_CLOUDY_WITH_SNOW_RAIN = 'HEAVILY_CLOUDY_WITH_SNOW_RAIN'

HEAVILY_CLOUDY_WITH_STRONG_RAIN = 'HEAVILY_CLOUDY_WITH_STRONG_RAIN'

HEAVILY_CLOUDY_WITH_THUNDER = 'HEAVILY_CLOUDY_WITH_THUNDER'

LIGHT_CLOUDY = 'LIGHT_CLOUDY'

STRONG_WIND = 'STRONG_WIND'

UNKNOWN = 'UNKNOWN'

class homematicip.base.enums.**WeatherDayTime**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

DAY = 'DAY'

NIGHT = 'NIGHT'

TWILIGHT = 'TWILIGHT'

class homematicip.base.enums.**WindValueType**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

AVERAGE_VALUE = 'AVERAGE_VALUE'

CURRENT_VALUE = 'CURRENT_VALUE'

MAX_VALUE = 'MAX_VALUE'

MIN_VALUE = 'MIN_VALUE'

class homematicip.base.enums.**WindowState**(**args*, ***kwds*)
Bases: *homematicip.base.enums.AutoNameEnum*

An enumeration.

CLOSED = 'CLOSED'

OPEN = 'OPEN'

TILTED = 'TILTED'

homematicip.base.functionalChannels module

class homematicip.base.functionalChannels.**AccelerationSensorChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the ACCELERATION_SENSOR_CHANNEL channel

accelerationSensorEventFilterPeriod = None
type: float

accelerationSensorMode = None
type: AccelerationSensorMode

accelerationSensorNeutralPosition = None
type: AccelerationSensorNeutralPosition

accelerationSensorSensitivity = None
type: AccelerationSensorSensitivity

accelerationSensorTriggerAngle = None
type: int

accelerationSensorTriggered = None
type: bool

from_json (*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

notificationSoundTypeHighToLow = None
type: NotificationSoundType

notificationSoundTypeLowToHigh = None
type: NotificationSoundType

class homematicip.base.functionalChannels.**AccessControllerChannel**
Bases: *homematicip.base.functionalChannels.DeviceBaseChannel*

this is the representative of the ACCESS_CONTROLLER_CHANNEL channel

from_json (*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**AlarmSirenChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the ALARM_SIREN_CHANNEL channel

class homematicip.base.functionalChannels.**AnalogOutputChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the ANALOG_OUTPUT_CHANNEL channel

analogOutputLevel = None
the analog output level (Volt?)

Type float

from_json (*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**AnalogRoomControlChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the ANALOG_ROOM_CONTROL_CHANNEL channel

from_json (*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**BlindChannel**
Bases: *homematicip.base.functionalChannels.ShutterChannel*

this is the representative of the BLIND_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**ChangeOverChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the CHANGE_OVER_CHANNEL channel

class homematicip.base.functionalChannels.**ClimateSensorChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the CLIMATE_SENSOR_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**ContactInterfaceChannel**
Bases: *homematicip.base.functionalChannels.ShutterContactChannel*

this is the representative of the CONTACT_INTERFACE_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**DehumidifierDemandChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the DEHUMIDIFIER_DEMAND_CHANNEL channel

class homematicip.base.functionalChannels.**DeviceBaseChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the DEVICE_BASE channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**DeviceBaseFloorHeatingChannel**
Bases: *homematicip.base.functionalChannels.DeviceBaseChannel*

this is the representative of the DEVICE_BASE_FLOOR_HEATING channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**DeviceGlobalPumpControlChannel**
Bases: *homematicip.base.functionalChannels.DeviceBaseChannel*

this is the representative of the DEVICE_GLOBAL_PUMP_CONTROL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**DeviceIncorrectPositionedChannel**
Bases: *homematicip.base.functionalChannels.DeviceBaseChannel*

this is the representative of the DEVICE_INCORRECT_POSITIONED channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**DeviceOperationLockChannel**
Bases: *homematicip.base.functionalChannels.DeviceBaseChannel*

this is the representative of the DEVICE_OPERATIONLOCK channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**DevicePermanentFullRxChannel**
Bases: *homematicip.base.functionalChannels.DeviceBaseChannel*

this is the representative of the DEVICE_PERMANENT_FULL_RX channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object

- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**DeviceRechargeableWithSabotage**
Bases: *homematicip.base.functionalChannels.DeviceSabotageChannel*

this is the representative of the DEVICE_RECHARGEABLE_WITH_SABOTAGE channel

badBatteryHealth = None
is the battery in a bad condition

Type bool

from_json (*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**DeviceSabotageChannel**
Bases: *homematicip.base.functionalChannels.DeviceBaseChannel*

this is the representative of the DEVICE_SABOTAGE channel

from_json (*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**DimmerChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the DIMMER_CHANNEL channel

from_json (*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**DoorChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the DoorChannel channel

from_json (*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**FloorTeminalBlockChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the FLOOR_TERMINAL_BLOCK_CHANNEL channel

class homematicip.base.functionalChannels.**FloorTerminalBlockLocalPumpChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the FLOOR_TERMINAL_BLOCK_LOCAL_PUMP_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**FloorTerminalBlockMechanicChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the class FLOOR_TERMINAL_BLOCK_MECHANIC_CHANNEL(FunctionalChannel) channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

valveState = None
the current valve state

Type *ValveState*

class homematicip.base.functionalChannels.**FunctionalChannel**
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

this is the base class for the functional channels

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**GenericInputChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the GENERIC_INPUT_CHANNEL channel

class homematicip.base.functionalChannels.**HeatDemandChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the HEAT_DEMAND_CHANNEL channel

class homematicip.base.functionalChannels.**HeatingThermostatChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the HEATING_THERMOSTAT_CHANNEL channel

automaticValveAdaptionNeeded = None
must the adaption re-run?

Type bool

from_json (*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

setPointTemperature = None
the current temperature which should be reached in the room

Type float

temperatureOffset = None
the offset temperature for the thermostat (+/- 3.5)

Type float

valveActualTemperature = None
the current measured temperature at the valve

Type float

valvePosition = None
the current position of the valve 0.0 = closed, 1.0 max opened

Type float

valveState = None
the current state of the valve

Type *ValveState*

class homematicip.base.functionalChannels.**InternalSwitchChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the INTERNAL_SWITCH_CHANNEL channel

from_json (*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**LightSensorChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the LIGHT_SENSOR_CHANNEL channel

averageIllumination = None
the average illumination value

Type float

currentIllumination = None
the current illumination value

Type float

from_json (*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

highestIllumination = None
the highest illumination value

Type float

lowestIllumination = None
the lowest illumination value

Type float

class homematicip.base.functionalChannels.**MainsFailureChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the MAINS_FAILURE_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**MotionDetectionChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the MOTION_DETECTION_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**MultiModeInputBlindChannel**
Bases: *homematicip.base.functionalChannels.BlindChannel*

this is the representative of the MULTI_MODE_INPUT_BLIND_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**MultiModeInputChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the MULTI_MODE_INPUT_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object

- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**MultiModeInputDimmerChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the MULTI_MODE_INPUT_DIMMER_CHANNEL channel

from_json (*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**MultiModeInputSwitchChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the MULTI_MODE_INPUT_SWITCH_CHANNEL channel

from_json (*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**NotificationLightChannel**
Bases: *homematicip.base.functionalChannels.DimmerChannel*

this is the representative of the NOTIFICATION_LIGHT_CHANNEL channel

from_json (*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

on = None
is the light turned on?

Type boolean

simpleRGBColorState = None
the color of the light

Type *RGBColorState*

class homematicip.base.functionalChannels.**PassageDetectorChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the PASSAGE_DETECTOR_CHANNEL channel

from_json (*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**PresenceDetectionChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the PRESENCE_DETECTION_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**RainDetectionChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the TILT_VIBRATION_SENSOR_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

rainSensorSensitivity = None
type: float

raining = None
type: bool

class homematicip.base.functionalChannels.**RotaryHandleChannel**
Bases: *homematicip.base.functionalChannels.ShutterContactChannel*

this is the representative of the ROTARY_HANDLE_CHANNEL channel

class homematicip.base.functionalChannels.**ShadingChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the SHADING_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**ShutterChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the SHUTTER_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**ShutterContactChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the SHUTTER_CONTACT_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**SingleKeyChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the SINGLE_KEY_CHANNEL channel

class homematicip.base.functionalChannels.**SmokeDetectorChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the SMOKE_DETECTOR_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**SwitchChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the SWITCH_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**SwitchMeasuringChannel**
Bases: *homematicip.base.functionalChannels.SwitchChannel*

this is the representative of the SWITCH_MEASURING_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**TemperaturDifferenceSensor2Channel**(*connection*)
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the TEMPERATURE_SENSOR_2_EXTERNAL_DELTA_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

temperatureExternalDelta = None
type: float

temperatureExternalOne = None
type: float

temperatureExternalTwo = None
type: float

class homematicip.base.functionalChannels.**TiltVibrationSensorChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the TILT_VIBRATION_SENSOR_CHANNEL channel

accelerationSensorEventFilterPeriod = None
type: float

accelerationSensorMode = None
type: AccelerationSensorMode

accelerationSensorSensitivity = None
type: AccelerationSensorSensitivity

accelerationSensorTriggerAngle = None
type: int

accelerationSensorTriggered = None
type: bool

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**WallMountedThermostatProChannel**
Bases: *homematicip.base.functionalChannels.WallMountedThermostatWithoutDisplayChannel*

this is the representative of the WALL_MOUNTED_THERMOSTAT_PRO_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**WallMountedThermostatWithoutDisplayChannel**
Bases: *homematicip.base.functionalChannels.ClimateSensorChannel*

this is the representative of the WALL_MOUNTED_THERMOSTAT_WITHOUT_DISPLAY_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**WaterSensorChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the WATER_SENSOR_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**WeatherSensorChannel**
Bases: *homematicip.base.functionalChannels.FunctionalChannel*

this is the representative of the WEATHER_SENSOR_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**WeatherSensorPlusChannel**
Bases: *homematicip.base.functionalChannels.WeatherSensorChannel*

this is the representative of the WEATHER_SENSOR_PLUS_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

class homematicip.base.functionalChannels.**WeatherSensorProChannel**
Bases: *homematicip.base.functionalChannels.WeatherSensorPlusChannel*

this is the representative of the WEATHER_SENSOR_PRO_CHANNEL channel

from_json(*js, groups: Iterable[homematicip.group.Group]*)
this function will load the functional channel object from a json object and the given groups

Parameters

- **js** (*dict*) – the json object
- **groups** (*Iterable[Group]*) – the groups for referencing

homematicip.base.helpers module

homematicip.base.helpers.**anonymizeConfig**(*config*, *pattern*, *format*, *flags=<RegexFlag.IGNORECASE: 2>*)

homematicip.base.helpers.**bytes2str**(*b*)

homematicip.base.helpers.**detect_encoding**(*b*)

homematicip.base.helpers.**get_functional_channel**(*channel_type*, *js*)

homematicip.base.helpers.**get_functional_channels**(*channel_type*, *js*)

homematicip.base.helpers.**handle_config**(*json_state: str*, *anonymize: bool*) → str

Module contents

3.1.2 Submodules

3.1.3 homematicip.EventHook module

class homematicip.EventHook.**EventHook**
Bases: object

fire(**args*, ***keywargs*)

3.1.4 homematicip.HomeMaticIPObject module

3.1.5 homematicip.auth module

class homematicip.auth.**Auth**(*home: homematicip.home.Home*)
Bases: object

confirmAuthToken(*authToken*)

connectionRequest(*access_point*, *devicename='homematicip-python'*) → requests.models.Response

isRequestAcknowledged()

requestAuthToken()

3.1.6 homematicip.class_maps module

3.1.7 homematicip.connection module

class homematicip.connection.**Connection**
Bases: *homematicip.base.base_connection.BaseConnection*

init(*accesspoint_id*, *lookup=True*, *lookup_url='https://lookup.homematic.com:48335/getHost'*, ***kwargs*)

3.1.8 homematicip.device module

class homematicip.device.**AccelerationSensor**(*connection*)
Bases: *homematicip.device.Device*

HMIP-SAM (Contact Interface flush-mount – 1 channel)

accelerationSensorEventFilterPeriod = None
type: float

accelerationSensorMode = None
type: AccelerationSensorMode

accelerationSensorNeutralPosition = None
type: AccelerationSensorNeutralPosition

accelerationSensorSensitivity = None
type: AccelerationSensorSensitivity

accelerationSensorTriggerAngle = None
type: int

accelerationSensorTriggered = None
type: bool

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

notificationSoundTypeHighToLow = None
type: NotificationSoundType

notificationSoundTypeLowToHigh = None
type: NotificationSoundType

set_acceleration_sensor_event_filter_period(*period: float*, *channelIndex=1*)

set_acceleration_sensor_mode(*mode: homematicip.base.enums.AccelerationSensorMode*, *channelIndex=1*)

set_acceleration_sensor_neutral_position(*neutralPosition: homematicip.base.enums.AccelerationSensorNeutralPosition*, *channelIndex=1*)

set_acceleration_sensor_sensitivity(*sensitivity: homematicip.base.enums.AccelerationSensorSensitivity*, *channelIndex=1*)

set_acceleration_sensor_trigger_angle(*angle: int*, *channelIndex=1*)

set_notification_sound_type(*soundType: homematicip.base.enums.NotificationSoundType*, *isHighToLow: bool*, *channelIndex=1*)

class homematicip.device.**AlarmSirenIndoor**(*connection*)
Bases: *homematicip.device.SabotageDevice*

HMIP-ASIR (Alarm Siren)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**AlarmSirenOutdoor**(*connection*)
Bases: *homematicip.device.AlarmSirenIndoor*

HMIP-ASIR-O (Alarm Siren Outdoor)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**Blind**(*connection*)
Bases: *homematicip.device.Shutter*

Base class for blind devices

set_slats_level(*slatsLevel=0.0*, *shutterLevel=None*, *channelIndex=1*)
sets the slats and shutter level

Parameters

- **slatsLevel** (*float*) – the new level of the slats. 0.0 = open, 1.0 = closed,
- **shutterLevel** (*float*) – the new level of the shutter. 0.0 = open, 1.0 = closed, None = use the current value
- **channelIndex** (*int*) – the channel to control

Returns the result of the _restCall

class homematicip.device.**BlindModule**(*connection*)
Bases: *homematicip.device.Device*

HMIP-HDM1 (Hunter Douglas & erfal window blinds)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_primary_shading_level(*primaryShadingLevel: float*)

set_secondary_shading_level(*primaryShadingLevel: float*, *secondaryShadingLevel: float*)

stop()
stops the current operation :returns: the result of the _restCall

class homematicip.device.**BrandBlind**(*connection*)
Bases: *homematicip.device.FullFlushBlind*

HMIP-BBL (Blind Actuator for brand switches)

class homematicip.device.**BrandDimmer**(*connection*)
Bases: *homematicip.device.Dimmer*

HMIP-BDT Brand Dimmer

class homematicip.device.**BrandPushButton**(*connection*)
Bases: *homematicip.device.PushButton*

HMIP-BRC2 (Remote Control for brand switches – 2x channels)

class homematicip.device.**BrandSwitchMeasuring**(*connection*)
Bases: *homematicip.device.SwitchMeasuring*

HMIP-BSM (Brand Switch and Meter)

class homematicip.device.**BrandSwitchNotificationLight**(*connection*)
Bases: *homematicip.device.Switch*

HMIP-BSL (Switch Actuator for brand switches – with signal lamp)

bottomLightChannelIndex = None
the channel number for the bottom light

Type int

set_rgb_dim_level(*channelIndex: int*, *rgb: homematicip.base.enums.RGBColorState*, *dimLevel: float*)
sets the color and dimlevel of the lamp

Parameters

- **channelIndex** (*int*) – the channelIndex of the lamp. Use self.topLightChannelIndex or self.bottomLightChannelIndex
- **rgb** (RGBColorState) – the color of the lamp
- **dimLevel** (*float*) – the dimLevel of the lamp. 0.0 = off, 1.0 = MAX

Returns the result of the _restCall

set_rgb_dim_level_with_time(*channelIndex: int*, *rgb: homematicip.base.enums.RGBColorState*, *dimLevel: float*, *onTime: float*, *rampTime: float*)

sets the color and dimlevel of the lamp

Parameters

- **channelIndex** (*int*) – the channelIndex of the lamp. Use self.topLightChannelIndex or self.bottomLightChannelIndex
- **rgb** (RGBColorState) – the color of the lamp
- **dimLevel** (*float*) – the dimLevel of the lamp. 0.0 = off, 1.0 = MAX
- **onTime** (*float*) –
- **rampTime** (*float*) –

Returns the result of the _restCall

topLightChannelIndex = None

the channel number for the top light

Type int

class homematicip.device.**ContactInterface**(*connection*)

Bases: *homematicip.device.SabotageDevice*

HMIP-SCI (Contact Interface Sensor)

from_json(*js*)

this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**Device**(*connection*)

Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

this class represents a generic homematic ip device

authorizeUpdate()

delete()

from_json(*js*)

this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

is_update_applicable()

load_functionalChannels(*groups: Iterable[homematicip.group.Group]*)

this function will load the functionalChannels into the device

set_label(*label*)

set_router_module_enabled(*enabled=True*)

class homematicip.device.**Dimmer**(*connection*)
Bases: *homematicip.device.Device*

Base dimmer device class

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_dim_level(*dimLevel=0.0*, *channelIndex=1*)

class homematicip.device.**DinRailBlind4**(*connection*)
Bases: *homematicip.device.Blind*

HmIP-DRBLI4 (Blind Actuator for DIN rail mount – 4 channels)

class homematicip.device.**DinRailDimmer3**(*connection*)
Bases: *homematicip.device.Dimmer*

HMIP-DRDI3 (Dimming Actuator Inbound 230V – 3x channels, 200W per channel) electrical DIN rail

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**DinRailSwitch**(*connection*)
Bases: *homematicip.device.FullFlushInputSwitch*

HMIP-DRSI1 (Switch Actuator for DIN rail mount – 1x channel)

class homematicip.device.**DinRailSwitch4**(*connection*)
Bases: *homematicip.device.Switch*

HMIP-DRSI4 (Homematic IP Switch Actuator for DIN rail mount – 4x channels)

class homematicip.device.**DoorModule**(*connection*)
Bases: *homematicip.device.Device*

Generic class for a door module

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

send_door_command(*doorCommand=<DoorCommand.STOP: 'STOP'>*)

class homematicip.device.**FloorTerminalBlock10**(*connection*)
Bases: *homematicip.device.FloorTerminalBlock6*

HMIP-FAL24-C10 (Floor Heating Actuator – 10x channels, 24V)

class homematicip.device.**FloorTerminalBlock12**(*connection*)
Bases: *homematicip.device.Device*

HMIP-FALMOT-C12 (Floor Heating Actuator – 12x channels, motorised)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_minimum_floor_heating_valve_position(*minimumFloorHeatingValvePosition: float*)
sets the minimum floot heating valve position

Parameters **minimumFloorHeatingValvePosition** (*float*) – the minimum valve position. must be between 0.0 and 1.0

Returns the result of the _restCall

class homematicip.device.**FloorTerminalBlock6**(*connection*)
Bases: *homematicip.device.Device*

HMIP-FAL230-C6 (Floor Heating Actuator - 6 channels, 230 V)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**FullFlushBlind**(*connection*)
Bases: *homematicip.device.FullFlushShutter*, *homematicip.device.Blind*

HMIP-FBL (Blind Actuator - flush-mount)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**FullFlushContactInterface**(*connection*)
Bases: *homematicip.device.Device*

HMIP-FCI1 (Contact Interface flush-mount – 1 channel)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**FullFlushContactInterface6**(*connection*)
Bases: *homematicip.device.Device*

HMIP-FCI6 (Contact Interface flush-mount – 6 channels)

class homematicip.device.**FullFlushDimmer**(*connection*)
Bases: *homematicip.device.Dimmer*

HMIP-FDT Dimming Actuator flush-mount

class homematicip.device.**FullFlushInputSwitch**(*connection*)
Bases: *homematicip.device.Switch*

HMIP-FSI16 (Switch Actuator with Push-button Input 230V, 16A)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**FullFlushShutter**(*connection*)
Bases: *homematicip.device.Shutter*

HMIP-FROLL (Shutter Actuator - flush-mount) / HMIP-BROLL (Shutter Actuator - Brand-mount)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**FullFlushSwitchMeasuring**(*connection*)
Bases: *homematicip.device.SwitchMeasuring*

HMIP-FSM, HMIP-FSM16 (Full flush Switch and Meter)

class homematicip.device.**GarageDoorModuleTormatic**(*connection*)
Bases: *homematicip.device.DoorModule*

HMIP-MOD-TM (Garage Door Module Tormatic)

class homematicip.device.**HeatingSwitch2**(*connection*)
Bases: *homematicip.device.Switch*

HMIP-WHS2 (Switch Actuator for heating systems – 2x channels)

class homematicip.device.**HeatingThermostat**(*connection*)
Bases: *homematicip.device.OperationLockableDevice*

HMIP-eTRV (Radiator Thermostat)

automaticValveAdaptionNeeded = None
must the adaption re-run?

Type bool

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

setPointTemperature = None
the current temperature which should be reached in the room

Type float

temperatureOffset = None
the offset temperature for the thermostat (+/- 3.5)

Type float

valveActualTemperature = None
the current measured temperature at the valve

Type float

valvePosition = None
the current position of the valve 0.0 = closed, 1.0 max opened

Type float

valveState = None
the current state of the valve

Type *ValveState*

class homematicip.device.**HeatingThermostatCompact**(*connection*)
Bases: *homematicip.device.SabotageDevice*

HMIP-eTRV-C (Heating-thermostat compact without display)

automaticValveAdaptionNeeded = None
must the adaption re-run?

Type bool

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

setPointTemperature = None
the current temperature which should be reached in the room

Type float

temperatureOffset = None
the offset temperature for the thermostat (+/- 3.5)

Type float

valveActualTemperature = None
the current measured temperature at the valve

Type float

valvePosition = None
the current position of the valve 0.0 = closed, 1.0 max opened

Type float

valveState = None
the current state of the valve

Type *ValveState*

class homematicip.device.**HeatingThermostatEvo**(*connection*)
Bases: *homematicip.device.OperationLockableDevice*

HMIP-eTRV-E (Heating-thermostat new evo version)

automaticValveAdaptionNeeded = None
must the adaption re-run?

Type bool

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

setPointTemperature = None
the current temperature which should be reached in the room

Type float

temperatureOffset = None
the offset temperature for the thermostat (+/- 3.5)

Type float

valveActualTemperature = None
the current measured temperature at the valve

Type float

valvePosition = None
the current position of the valve 0.0 = closed, 1.0 max opened

Type float

valveState = None
the current state of the valve

Type *ValveState*

class homematicip.device.**HoermannDrivesModule**(*connection*)
Bases: *homematicip.device.DoorModule*

HMIP-MOD-HO (Garage Door Module for Hörmann)

class homematicip.device.**HomeControlAccessPoint**(*connection*)
Bases: *homematicip.device.Device*

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**KeyRemoteControl4**(*connection*)
Bases: *homematicip.device.PushButton*

HMIP-KRC4 (Key Ring Remote Control - 4 buttons)

class homematicip.device.**KeyRemoteControlAlarm**(*connection*)
Bases: *homematicip.device.Device*

HMIP-KRCA (Key Ring Remote Control - alarm)

class homematicip.device.**LightSensor**(*connection*)
Bases: *homematicip.device.Device*

HMIP-SLO (Light Sensor outdoor)

averageIllumination = None
the average illumination value

Type float

currentIllumination = None
the current illumination value

Type float

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

highestIllumination = None
the highest illumination value

Type float

lowestIllumination = None
the lowest illumination value

Type float

class homematicip.device.**MotionDetectorIndoor**(*connection*)
Bases: *homematicip.device.SabotageDevice*

HMIP-SMI (Motion Detector with Brightness Sensor - indoor)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**MotionDetectorOutdoor**(*connection*)
Bases: *homematicip.device.Device*

HMIP-SMO-A (Motion Detector with Brightness Sensor - outdoor)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**MotionDetectorPushButton**(*connection*)
Bases: *homematicip.device.MotionDetectorOutdoor*

HMIP-SMI55 (Motion Detector with Brightness Sensor and Remote Control - 2-button)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**MultiIOBox**(*connection*)
Bases: *homematicip.device.Switch*

HMIP-MIOB (Multi IO Box for floor heating & cooling)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**OpenCollector8Module**(*connection*)
Bases: *homematicip.device.Switch*

HMIP-MOD-OC8 (Open Collector Module)

class homematicip.device.**OperationLockableDevice**(*connection*)
Bases: *homematicip.device.Device*

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_operation_lock(*operationLock=True*)

class homematicip.device.**PassageDetector**(*connection*)
Bases: *homematicip.device.SabotageDevice*

HMIP-SPDR (Passage Detector)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**PlugableSwitch**(*connection*)
Bases: *homematicip.device.Switch*

HMIP-PS (Pluggable Switch), HMIP-PCBS (Switch Circuit Board - 1 channel)

class homematicip.device.**PlugableSwitchMeasuring**(*connection*)
Bases: *homematicip.device.SwitchMeasuring*

HMIP-PSM (Pluggable Switch and Meter)

class homematicip.device.**PluggableDimmer**(*connection*)
Bases: *homematicip.device.Dimmer*

HMIP-PDT Pluggable Dimmer

class homematicip.device.**PluggableMainsFailureSurveillance**(*connection*)
Bases: *homematicip.device.Device*

HMIP-PMFS (Plugable Power Supply Monitoring)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**PresenceDetectorIndoor**(*connection*)
Bases: *homematicip.device.SabotageDevice*

HMIP-SPI (Presence Sensor - indoor)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**PrintedCircuitBoardSwitch2**(*connection*)
Bases: *homematicip.device.Switch*

HMIP-PCBS2 (Switch Circuit Board - 2x channels)

class homematicip.device.**PrintedCircuitBoardSwitchBattery**(*connection*)
Bases: *homematicip.device.Switch*

HMIP-PCBS-BAT (Printed Circuit Board Switch Battery)

class homematicip.device.**PushButton**(*connection*)
Bases: *homematicip.device.Device*

HMIP-WRC2 (Wall-mount Remote Control - 2-button)

class homematicip.device.**PushButton6**(*connection*)
Bases: *homematicip.device.PushButton*

HMIP-WRC6 (Wall-mount Remote Control - 6-button)

class homematicip.device.**PushButtonFlat**(*connection*)
Bases: *homematicip.device.PushButton*

HmIP-WRCC2 (Wall-mount Remote Control – flat)

class homematicip.device.**RainSensor**(*connection*)
Bases: *homematicip.device.Device*

HMIP-SRD (Rain Sensor)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

rainSensorSensitivity = None
type: float

raining = None
type: bool

class homematicip.device.**RemoteControl8**(*connection*)
Bases: *homematicip.device.PushButton*

HMIP-RC8 (Remote Control - 8 buttons)

class homematicip.device.**RemoteControl8Module**(*connection*)
Bases: *homematicip.device.RemoteControl8*

HMIP-MOD-RC8 (Open Collector Module Sender - 8x)

class homematicip.device.**RoomControlDevice**(*connection*)
Bases: *homematicip.device.WallMountedThermostatPro*

ALPHA-IP-RBG (Alpha IP Wall Thermostat Display)

class homematicip.device.**RoomControlDeviceAnalog**(*connection*)
Bases: *homematicip.device.Device*

ALPHA-IP-RBGa (ALpha IP Wall Thermostat Display analog)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**RotaryHandleSensor**(*connection*)
Bases: *homematicip.device.SabotageDevice*

HMIP-SRH

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**SabotageDevice**(*connection*)
Bases: *homematicip.device.Device*

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**Shutter**(*connection*)
Bases: *homematicip.device.Device*

Base class for shutter devices

set_shutter_level(*level=0.0*, *channelIndex=1*)
sets the shutter level

Parameters

- **level** (*float*) – the new level of the shutter. 0.0 = open, 1.0 = closed
- **channelIndex** (*int*) – the channel to control

Returns the result of the _restCall

set_shutter_stop(*channelIndex=1*)
stops the current shutter operation

Parameters **channelIndex** (*int*) – the channel to control

Returns the result of the _restCall

class homematicip.device.**ShutterContact**(*connection*)
Bases: *homematicip.device.SabotageDevice*

HMIP-SWDO (Door / Window Contact - optical) / HMIP-SWDO-I (Door / Window Contact Invisible - optical)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**ShutterContactMagnetic**(*connection*)
Bases: *homematicip.device.Device*

HMIP-SWDM / HMIP-SWDM-B2 (Door / Window Contact - magnetic)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**ShutterContactOpticalPlus**(*connection*)
Bases: *homematicip.device.ShutterContact*

HmIP-SWDO-PL (Window / Door Contact – optical, plus)

class homematicip.device.**SmokeDetector**(*connection*)
Bases: *homematicip.device.Device*

HMIP-SWSD (Smoke Alarm with Q label)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**Switch**(*connection*)
Bases: *homematicip.device.Device*

Generic Switch class

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_switch_state(*on=True*, *channelIndex=1*)

turn_off(*channelIndex=1*)

turn_on(*channelIndex=1*)

class homematicip.device.**SwitchMeasuring**(*connection*)
Bases: *homematicip.device.Switch*

Generic class for Switch and Meter

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

reset_energy_counter()

class homematicip.device.**TemperaturDifferenceSensor2**(*connection*)
Bases: *homematicip.device.Device*

HmIP-STE2-PCB (Temperature Difference Sensors - 2x sensors)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

temperatureExternalDelta = None
type: float

temperatureExternalOne = None
type: float

temperatureExternalTwo = None
type: float

class homematicip.device.**TemperatureHumiditySensorDisplay**(*connection*)
Bases: *homematicip.device.Device*

HMIP-STHD (Temperature and Humidity Sensor with display - indoor)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_display(*display: homematicip.base.enums.ClimateControlDisplay = <ClimateControlDisplay.ACTUAL: 'ACTUAL'>*)

class homematicip.device.**TemperatureHumiditySensorOutdoor**(*connection*)
Bases: *homematicip.device.Device*

HMIP-STHO (Temperature and Humidity Sensor outdoor)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**TemperatureHumiditySensorWithoutDisplay**(*connection*)
Bases: *homematicip.device.Device*

HMIP-STH (Temperature and Humidity Sensor without display - indoor)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**TiltVibrationSensor**(*connection*)
Bases: *homematicip.device.Device*

HMIP-STV (Inclination and vibration Sensor)

accelerationSensorEventFilterPeriod = None
type: float

accelerationSensorMode = None
type: AccelerationSensorMode

accelerationSensorSensitivity = None
type: AccelerationSensorSensitivity

accelerationSensorTriggerAngle = None
type: int

accelerationSensorTriggered = None
type: bool

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_acceleration_sensor_event_filter_period(*period: float*, *channelIndex=1*)

set_acceleration_sensor_mode(*mode: homematicip.base.enums.AccelerationSensorMode*, *channelIndex=1*)

set_acceleration_sensor_sensitivity(*sensitivity: homematicip.base.enums.AccelerationSensorSensitivity*, *channelIndex=1*)

set_acceleration_sensor_trigger_angle(*angle: int*, *channelIndex=1*)

class homematicip.device.**WallMountedThermostatBasicHumidity**(*connection*)
Bases: *homematicip.device.WallMountedThermostatPro*

HMIP-WTH-B (Wall Thermostat – basic)

class homematicip.device.**WallMountedThermostatPro**(*connection*)
Bases: *homematicip.device.TemperatureHumiditySensorDisplay*, *homematicip.device.OperationLockableDevice*

HMIP-WTH, HMIP-WTH-2 (Wall Thermostat with Humidity Sensor) / HMIP-BWTH (Brand Wall Thermostat with Humidity Sensor)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**WaterSensor**(*connection*)
Bases: *homematicip.device.Device*

HMIP-SWD (Water Sensor)

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_acoustic_alarm_signal(*acousticAlarmSignal: homematicip.base.enums.AcousticAlarmSignal*)

set_acoustic_alarm_timing(*acousticAlarmTiming: homematicip.base.enums.AcousticAlarmTiming*)

set_acoustic_water_alarm_trigger(*acousticWaterAlarmTrigger: homematicip.base.enums.WaterAlarmTrigger*)

set_inapp_water_alarm_trigger(*inAppWaterAlarmTrigger: homematicip.base.enums.WaterAlarmTrigger*)

set_siren_water_alarm_trigger(*sirenWaterAlarmTrigger: homematicip.base.enums.WaterAlarmTrigger*)

class homematicip.device.**WeatherSensor**(*connection*)
Bases: *homematicip.device.Device*

HMIP-SWO-B

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**WeatherSensorPlus**(*connection*)
Bases: *homematicip.device.Device*

HMIP-SWO-PL

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**WeatherSensorPro**(*connection*)
Bases: *homematicip.device.Device*

HMIP-SWO-PR

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.device.**WiredDimmer3**(*connection*)
Bases: *homematicip.device.Dimmer*

HMIPW-DRD3 (Homematic IP Wired Dimming Actuator – 3x channels)

class homematicip.device.**WiredInput32**(*connection*)
Bases: *homematicip.device.FullFlushContactInterface*

HMIPW-DRI32 (Homematic IP Wired Inbound module – 32x channels)

class homematicip.device.**WiredSwitch8**(*connection*)
Bases: *homematicip.device.Switch*

HMIPW-DRS8 (Homematic IP Wired Switch Actuator – 8x channels)

3.1.9 homematicip.functionalHomes module

class homematicip.functionalHomes.**AccessControlHome**(*connection*)
Bases: *homematicip.functionalHomes.FunctionalHome*

from_json(*js*, *groups: List[homematicip.group.Group]*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.functionalHomes.**FunctionalHome**(*connection*)
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

assignGroups(*gids*, *groups: List[homematicip.group.Group]*)

from_json(*js*, *groups: List[homematicip.group.Group]*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.functionalHomes.**IndoorClimateHome**(*connection*)
Bases: *homematicip.functionalHomes.FunctionalHome*

from_json(*js*, *groups: List[homematicip.group.Group]*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.functionalHomes.**LightAndShadowHome**(*connection*)
Bases: *homematicip.functionalHomes.FunctionalHome*

from_json(*js*, *groups: List[homematicip.group.Group]*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.functionalHomes.**SecurityAndAlarmHome**(*connection*)
Bases: *homematicip.functionalHomes.FunctionalHome*

from_json(*js, groups: List[homematicip.group.Group]*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.functionalHomes.**WeatherAndEnvironmentHome**(*connection*)
Bases: *homematicip.functionalHomes.FunctionalHome*

3.1.10 homematicip.group module

class homematicip.group.**AlarmSwitchingGroup**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_on_time(*onTimeSeconds*)

set_signal_acoustic(*signalAcoustic=<AcousticAlarmSignal.FREQUENCY_FALLING: 'FREQUENCY_FALLING'>*)

set_signal_optical(*signalOptical=<OpticalAlarmSignal.BLINKING_ALTERNATELY_REPEATING: 'BLINKING_ALTERNATELY_REPEATING'>*)

test_signal_acoustic(*signalAcoustic=<AcousticAlarmSignal.FREQUENCY_FALLING: 'FREQUENCY_FALLING'>*)

test_signal_optical(*signalOptical=<OpticalAlarmSignal.BLINKING_ALTERNATELY_REPEATING: 'BLINKING_ALTERNATELY_REPEATING'>*)

class homematicip.group.**EnvironmentGroup**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.group.**ExtendedLinkedShutterGroup**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_shutter_level(*level*)

set_shutter_stop()

set_slats_level(*slatsLevel=0.0*, *shutterLevel=None*)

class homematicip.group.**ExtendedLinkedSwitchingGroup**(*connection*)
Bases: *homematicip.group.SwitchGroupBase*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_on_time(*onTimeSeconds*)

class homematicip.group.**Group**(*connection*)
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

this class represents a group

delete()

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_label(*label*)

class homematicip.group.**HeatingChangeoverGroup**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.group.**HeatingCoolingDemandBoilerGroup**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.group.**HeatingCoolingDemandGroup**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.group.**HeatingCoolingDemandPumpGroup**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.group.**HeatingCoolingPeriod**(*connection*)
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.group.**HeatingCoolingProfile**(*connection*)
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

get_details()

update_profile()

class homematicip.group.**HeatingCoolingProfileDay**(*connection*)
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.group.**HeatingDehumidifierGroup**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.group.**HeatingExternalClockGroup**(*connection*)
Bases: *homematicip.group.Group*

class homematicip.group.**HeatingFailureAlertRuleGroup**(*connection*)
Bases: *homematicip.group.Group*

checkInterval = None
how often the system will check for an error

Type int

enabled = None
is this rule active

Type bool

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

heatingFailureValidationResult = None
the heating failure value

Type *HeatingFailureValidationType*

lastExecutionTimestamp = None
last time of execution

Type datetime

validationTimeout = None
time in ms for the validation period. default 24Hours

Type int

class homematicip.group.**HeatingGroup**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_active_profile(*index*)

set_boost(*enable=True*)

set_boost_duration(*duration: int*)

set_control_mode(*mode=<ClimateControlMode.AUTOMATIC: 'AUTOMATIC'>*)

set_point_temperature(*temperature*)

class homematicip.group.**HeatingHumidyLimiterGroup**(*connection*)
Bases: *homematicip.group.Group*

class homematicip.group.**HeatingTemperatureLimiterGroup**(*connection*)
Bases: *homematicip.group.Group*

class homematicip.group.**HotWaterGroup**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_profile_mode(*profileMode: homematicip.base.enums.ProfileMode*)

class homematicip.group.**HumidityWarningRuleGroup**(*connection*)
Bases: *homematicip.group.Group*

enabled = None
is this rule active

Type bool

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

humidityLowerThreshold = None
the lower humidity threshold

Type int

humidityUpperThreshold = None
the upper humidity threshold

Type int

humidityValidationResult = None
the current humidity result

Type *HumidityValidationType*

lastExecutionTimestamp = None
last time of execution

Type datetime

lastStatusUpdate = None
last time the humidity got updated

Type datetime

outdoorClimateSensor = None
the climate sensor which get used as an outside reference. None if OpenWeatherMap will be used

Type *Device*

triggered = None
is it currently triggered?

Type bool

ventilationRecommended = None
should the windows be opened?

Type bool

class homematicip.group.**InboxGroup**(*connection*)
Bases: *homematicip.group.Group*

class homematicip.group.**LinkedSwitchingGroup**(*connection*)
Bases: *homematicip.group.Group*

set_light_group_switches(*devices*)

class homematicip.group.**LockOutProtectionRule**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.group.**MetaGroup**(*connection*)
Bases: *homematicip.group.Group*

a meta group is a "Room" inside the homematic configuration

from_json(*js*, *devices*, *groups*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.group.**OverHeatProtectionRule**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.group.**SecurityGroup**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.group.**SecurityZoneGroup**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.group.**ShutterProfile**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_profile_mode(*profileMode: homematicip.base.enums.ProfileMode*)

set_shutter_level(*level*)

set_shutter_stop()

set_slats_level(*slatsLevel*, *shutterlevel*)

class homematicip.group.**ShutterWindProtectionRule**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.group.**SmokeAlarmDetectionRule**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.group.**SwitchGroupBase**(*connection*)
Bases: *homematicip.group.Group*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_switch_state(*on=True*)

turn_off()

turn_on()

class homematicip.group.**SwitchingGroup**(*connection*)
Bases: *homematicip.group.SwitchGroupBase*

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_shutter_level(*level*)

set_shutter_stop()

set_slats_level(*slatsLevel*, *shutterlevel*)

class homematicip.group.**SwitchingProfileGroup**(*connection*)
Bases: *homematicip.group.Group*

create(*label*)

from_json(*js*, *devices*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_group_channels()

set_profile_mode(*devices*, *automatic=True*)

class homematicip.group.**TimeProfile**(*connection*)
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

get_details()

class homematicip.group.**TimeProfilePeriod**(*connection*)
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

3.1.11 homematicip.home module

class homematicip.home.**AccessPointUpdateState**(*connection*)
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.home.**Client**(*connection*)
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

A client is an app which has access to the access point. e.g. smartphone, 3th party apps, google home, conrad connect

c2cServiceIdentifier = None
the c2c service name

Type str

clientType = None
the type of this client

Type *ClientType*

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

homeId = None
the home where the client belongs to

Type str

id = None
the unique id of the client

Type str

label = None
a human understandable name of the client

Type str

class homematicip.home.**Home**(*connection=None*)
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

this class represents the 'Home' of the homematic ip

accessPointUpdateStates = None
a map of all access points and their updateStates

Type Map

activate_absence_permanent()
activates the absence forever

activate_absence_with_duration(*duration: int*)
activates the absence mode for a given time

Parameters **duration** (*int*) – the absence duration in minutes

activate_absence_with_period(*endtime: datetime.datetime*)
activates the absence mode until the given time

Parameters **endtime** (*datetime*) – the time when the absence should automatically be disabled

activate_vacation(*endtime: datetime.datetime*, *temperature: float*)
activates the vatation mode until the given time

Parameters

- **endtime** (*datetime*) – the time when the vatation mode should automatically be disabled
- **temperature** (*float*) – the settemperature during the vacation mode

clients = None
a collection of all clients in home

Type List[*Client*]

currentAPVersion = None
the current version of the access point

Type str

deactivate_absence()
deactivates the absence mode immediately

deactivate_vacation()
deactivates the vacation mode immediately

delete_group(*group: homematicip.group.Group*)
deletes the given group from the cloud

Parameters **group** (Group) – the group to delete

devices = None
a collection of all devices in home

Type List[*Device*]

disable_events()

download_configuration() → str
downloads the current configuration from the cloud

Returns the downloaded configuration or an errorCode

enable_events(*enable_trace=False*, *ping_interval=20*)

fire_create_event(**args*, ***kwargs*)
Trigger the method tied to _on_create

from_json(*js_home*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

functionalHomes = None
a collection of all functionalHomes in the home

get_OAuth_OTK()

get_current_state(*clearConfig: bool = False*)
downloads the current configuration and parses it into self

Parameters

- **clearConfig** (*bool*) – if set to true, this function will remove all old objects
- **self.devices, self.client, .. to have a fresh config instead of reparsing them** (*from*) –

get_functionalHome(*functionalHomeType: type*) → homematicip.functionalHomes.FunctionalHome
gets the specified functionalHome

Parameters **functionalHome** (*type*) – the type of the functionalHome which should be returned

Returns the FunctionalHome or None if it couldn't be found

get_security_journal()

get_security_zones_activation() -> (<class 'bool'>, <class 'bool'>)
returns the value of the security zones if they are armed or not

Returns

internal True if the internal zone is armed

external True if the external zone is armed

groups = None
a collection of all groups in the home

Type List[*Group*]

id = None
the SGTIN of the access point

Type str

init(*access_point_id*, *lookup=True*)

location = None
the location of the AP

Type *Location*

on_create(*handler*)
Adds an event handler to the create method. Fires when a device is created.

pinAssigned = None
determines if a pin is set on this access point

Type bool

remove_callback(*handler*)
Remove event handler.

rules = None
a collection of all rules in the home

Type List[*Rule*]

search_client_by_id(*clientID*) → homematicip.home.Client
searches a client by given id

Parameters **clientID** (*str*) – the client to search for

Returns the client object or None if it couldn't find a client

search_device_by_id(*deviceID*) → homematicip.device.Device
searches a device by given id

Parameters **deviceID** (*str*) – the device to search for

Returns the Device object or None if it couldn't find a device

search_group_by_id(*groupID*) → homematicip.group.Group
searches a group by given id

Parameters **groupID** (*str*) – groupID the group to search for

Returns the group object or None if it couldn't find a group

search_rule_by_id(*ruleID*) → homematicip.rule.Rule
searches a rule by given id

Parameters **ruleID** (*str*) – the rule to search for

Returns the rule object or None if it couldn't find a rule

set_auth_token(*auth_token*)

set_intrusion_alert_through_smoke_detectors(*activate: bool = True*)
activate or deactivate if smoke detectors should "ring" during an alarm

Parameters **activate** (*bool*) – True will let the smoke detectors "ring" during an alarm

set_location(*city*, *latitude*, *longitude*)

set_pin(*newPin: str*, *oldPin: str = None*) → dict
sets a new pin for the home

Parameters

- **newPin** (*str*) – the new pin
- **oldPin** (*str*) – optional, if there is currently a pin active it must be given here. Otherwise it will not be possible to set the new pin

Returns the result of the call

set_powermeter_unit_price(*price*)

set_security_zones_activation(*internal=True*, *external=True*)
this function will set the alarm system to armed or disable it

Parameters

- **internal** (*bool*) – activates/deactivates the internal zone
- **external** (*bool*) – activates/deactivates the external zone

Examples

arming while being at home

```
>>> home.set_security_zones_activation(False,True)
```

arming without being at home

```
>>> home.set_security_zones_activation(True,True)
```

disarming the alarm system

```
>>> home.set_security_zones_activation(False,False)
```

set_timezone(*timezone: str*)
sets the timezone for the AP. e.g. "Europe/Berlin" :param timezone: the new timezone :type timezone: str

set_zone_activation_delay(*delay*)

set_zones_device_assignment(*internal_devices*, *external_devices*) → dict
sets the devices for the security zones :param internal_devices: the devices which should be used for the internal zone :type internal_devices: List[Device] :param external_devices: the devices which should be used for the external(hull) zone :type external_devices: List[Device]

Returns the result of _restCall

start_inclusion(*deviceId*)
start inclusion mode for specific device :param deviceId: sgtin of device

update_home(*json_state*, *clearConfig: bool = False*)
parse a given json configuration into self. This will update the whole home including devices, clients and groups.

Parameters

- **clearConfig** (*bool*) – if set to true, this function will remove all old objects
- **self.devices, self.client, .. to have a fresh config instead of reparsing them** (*from*) –

update_home_only(*js_home*, *clearConfig: bool = False*)
parse a given home json configuration into self. This will update only the home without updating devices, clients and groups.

Parameters

- **clearConfig** (*bool*) – if set to true, this function will remove all old objects
- **self.devices, self.client, .. to have a fresh config instead of reparsing them** (*from*) –

weather = None
the current weather

Type *Weather*

websocket_reconnect_on_error = None
switch to enable/disable automatic reconnection of the websocket (default=True)

Type bool

class homematicip.home.**Location**(*connection*)
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

This class represents the possible location

city = None
the name of the city

Type str

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

latitude = None
the latitude of the location

Type float

longitude = None
the longitue of the location

Type float

class homematicip.home.**OAuthOTK**(*connection*)
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.home.**Weather**(*connection*)
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

this class represents the weather of the home location

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

humidity = None
the current humidity

Type float

maxTemperature = None
the maximum temperature of the day

Type float

minTemperature = None
the minimum temperature of the day

Type float

temperature = None
the current temperature

Type float

vaporAmount = None
the current vapor

Type float

weatherCondition = None
the current weather

Type *WeatherCondition*

weatherDayTime = None
the current datime

Type datetime

windDirection = None
the current wind direction in 360° where 0° is north

Type int

windSpeed = None
the current windspeed

Type float

3.1.12 homematicip.rule module

class homematicip.rule.**Rule**(*connection*)
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

this class represents the automation rule

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

set_label(*label*)
sets the label of the rule

class homematicip.rule.**SimpleRule**(*connection*)
Bases: *homematicip.rule.Rule*

This class represents a "Simple" automation rule

disable()
disables the rule

enable()
enables the rule

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

get_simple_rule()

set_rule_enabled_state(*enabled*)
enables/disables this rule

3.1.13 homematicip.securityEvent module

class homematicip.securityEvent.**AccessPointConnectedEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityEvent*

class homematicip.securityEvent.**AccessPointDisconnectedEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityEvent*

class homematicip.securityEvent.**ActivationChangedEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityZoneEvent*

class homematicip.securityEvent.**ExternalTriggeredEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityEvent*

class homematicip.securityEvent.**MainsFailureEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityEvent*

class homematicip.securityEvent.**MoistureDetectionEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityEvent*

class homematicip.securityEvent.**OfflineAlarmEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityEvent*

class homematicip.securityEvent.**OfflineWaterDetectionEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityEvent*

class homematicip.securityEvent.**SabotageEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityEvent*

class homematicip.securityEvent.**SecurityEvent**(*connection*)
Bases: homematicip.base.HomeMaticIPObject.HomeMaticIPObject

this class represents a security event

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.securityEvent.**SecurityZoneEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityEvent*

This class will be used by other events which are just adding "securityZoneValues"

from_json(*js*)
this method will parse the homematicip object from a json object

Parameters **js** – the json object to parse

class homematicip.securityEvent.**SensorEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityEvent*

class homematicip.securityEvent.**SilenceChangedEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityZoneEvent*

class homematicip.securityEvent.**SmokeAlarmEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityEvent*

class homematicip.securityEvent.**WaterDetectionEvent**(*connection*)
Bases: *homematicip.securityEvent.SecurityEvent*

3.1.14 Module contents

class homematicip.**HmipConfig**(*auth_token*, *access_point*, *log_level*, *log_file*, *raw_config*)
Bases: tuple

access_point
Alias for field number 1

auth_token
Alias for field number 0

log_file
Alias for field number 3

log_level
Alias for field number 2

raw_config
Alias for field number 4

homematicip.**find_and_load_config_file**() → homematicip.HmipConfig

homematicip.**get_config_file_locations**() → []

homematicip.**load_config_file**(*config_file: str*) → homematicip.HmipConfig
Loads the config ini file. :raises a FileNotFoundError when the config file does not exist.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

h

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W

Y