heatzy.

Heatzy API

Table of Contents

Informations about the user

Heatzy_Application_Id : c70a66ff039d41b4a220e198b0fcc8b3
Product_Key Heatzy_Pilote : 9420ae048da545c88fc6274d204dd25f
Product_Key duo Flam / Plugzy : f71ee820660f4f358db8b8a474689726
Product_Key Pilote_Gen2 : 51d16c22a5f74280bc3cfe9ebcdc6402
Product_Key Inea : fc89066ee74c4149a9beb37d4ea93604

X-Gizwits-User-token is the token given when register. You can obtain it by using GET /user/login

X-Gizwits-Timestamp is the timestamp at the moment you are giving the order and should not be more than 5 minutes different from the server timestamp.

X-Gizwits-Signature : the calculation method is lower (md5(product_secret + timestamp)), timestamp need to be the same that X-Gizwits-Timestamp. Product_secret is not given in this document. If needed, please contact us.

Programmation :

There is two way using programmation on new category of products (PILOTE second generation, FLAM and INEA), You can both use scheduler to implement the programmation on the server or store it directly in the device using datapoint.

If you want to use datapoint, each data represent 2 hours. Each 30 min is coded using 2 bits :

- 00 : comfort mode
- 01 : eco mode
- 10 : anti-freeze mode

The advantage using datapoint is that after set-up, product will not need Wifi anymore.

Binding management

POST /app/bind_mac

Bind the device according to the MAC address

→ **Header**
'Accept: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'
'X-Gizwits-User-token: **User_Token**'
'X-Gizwits-Timestamp: **Timestamp**'
'X-Gizwits-Signature: **Gizwits-Signature**'

→ **Body**
{
"product_key": "string",
"mac": "string"
}

→ **URL**
https://euapi.gizwits.com/app/bind_mac

→ **Curl Command**
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'X-Gizwits-Timestamp: **Timestamp**' --header 'X-Gizwits-Signature: **Gizwits-Signature**' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**' --header 'X-Gizwits-User-token: **User_Token**' -d '{"product_key": "string", "mac": "string"}' 'https://euapi.gizwits.com/app/bind_mac'

→ **Example returned value**
{
"product_key": "string",
"did": "string",
"mac": "string",
"is_online": true,
"passcode": "string",
"host": "string",
"port": 0,
"port_s": 0,
"ws_port": 0,
"wss_port": 0,
"remark": "string",
"is_disabled": true,
"type": "string",

"dev_alias": "string",
"proto_ver": "01"
}

DELETE /app/bindings

Unbind a product

→ **Header**
'Accept: application/json'
'Content-Type: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'
'X-Gizwits-User-token: **User_Token**'

→ **Body**
{
"devices": [
{
"did": "string"
}
]
}

→ **URL**
https://euapi.gizwits.com/app/bindings

→ **Curl Command**
curl -X DELETE --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'X-Gizwits-User-token: **User_Token**' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**' -d '{ "devices": [{ "did": "string" }] }' 'https://euapi.gizwits.com/app/bindings'

GET /app/bindings

Get the binding list

→ **Header**
'Accept: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'
'X-Gizwits-User-token: **User_Token**'

→ **URL**
https://euapi.gizwits.com/app/bindings?limit=20&skip=0

→ **Curl Command**

curl -X GET --header 'Accept: application/json' --header 'X-Gizwits-User-token: **User_Token**' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**' 'https://euapi.gizwits.com/app/bindings?limit=20&skip=0'

→ **Example returned value**

{
"devices": [
{
"product_key": "string",
"did": "string",
"mac": "string",
"is_online": true,
"passcode": "string",
"host": "string",
"port": 0,
"port_s": 0,
"ws_port": 0,
"wss_port": 0,
"remark": "string",
"is_disabled": true,
"type": "string",
"dev_alias": "string",
"proto_ver": "01"
}
]
}

PUT /app/bindings/{did}

Modify the binding information

→ **Header**

'Accept: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'
'X-Gizwits-User-token: **User_Token**'

→ **Body**

{
"remark": "string",
"dev_alias": "string"
}

→ **URL**

https://euapi.gizwits.com/app/bindings/**{did}**

→ **Curl Command**

curl -X PUT --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'X-Gizwits-User-token: **User_Token**' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**' -d '{ "remark": "string", "dev_alias": "string" }' 'https://euapi.gizwits.com/app/bindings/**{did}**'

Timing task management

GET /app/devices/{did}/scheduler

Get timing task

⟶ Header

'Accept: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'
'X-Gizwits-User-token: **User_Token**'

⟶ URL

https://euapi.gizwits.com/app/devices/**{did}**/scheduler?limit=20&skip=0

⟶ Curl Command

curl -X GET --header 'Accept: application/json' --header 'X-Gizwits-User-token: **User_Token**' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**' 'https://euapi.gizwits.com/app/devices/**{did}**/scheduler?limit=20&skip=0'

⟶ Example returned value

[
{
"attrs": {},
"date": "2017-03-08",
"time": "string",
"repeat": "string",
"days": [
0
],
"start_date": "2017-03-08",
"end_date": "2017-03-08",
"enabled": true,
"remark": "string",
"id": "string",
"created_at": "string"
}
]

POST /app/devices/{did}/scheduler

Create a scheduler task

⟶ Header

'Accept: application/json'
'Content-Type: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'
'X-Gizwits-User-token: **User_Token**'

⟶ Body

```
{
"attrs": {},
"date": "2017-03-10",
"time": "string",
"repeat": "string",
"days": [
0
],
"start_date": "2017-03-10",
"end_date": "2017-03-10",
"enabled": true,
"remark": "string"
}
```

⟶ URL

https://euapi.gizwits.com/app/devices/**{did}**/scheduler

⟶ Curl Command

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept:
application/json' --header 'X-Gizwits-User-token: User_Token' --header
'X-Gizwits-Application-Id: Heatzy_Application_Id' -d '{ "attrs": {}, "date":
"2017-03-08", "time": "string", "repeat": "string", "days": [ 0 ], "start_date":
"2017-03-08", "end_date": "2017-03-08", "enabled": true, "remark": "string" }'
'https://euapi.gizwits.com/app/devices/{did}/scheduler'
```

⟶ Exemple returned value

```
{
"id": "string"
}
```

DELETE /app/devices/{did}/scheduler/{id}

⟶ **Header**
'Accept: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'
'X-Gizwits-User-token: **User_Token**'

⟶ **URL**
https://euapi.gizwits.com/app/devices/**{did}**/scheduler/**{id}**

⟶ **Curl Command**
curl -X DELETE --header 'Accept: application/json' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**' --header 'X-Gizwits-User-token: **User_Token**' 'https://euapi.gizwits.com/app/devices/**{did}**/scheduler/**{id}**'

PUT /app/devices/{did}/scheduler/{id}

⟶ **Header**
'Accept: application/json'
'Content-Type: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'
'X-Gizwits-User-token: **User_Token**'

⟶ **URL**
'https://euapi.gizwits.com/app/devices/**{did}**/scheduler/**{id}**'

⟶ **Curl Command**
curl -X PUT --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**' --header 'X-Gizwits-User-token: **User_Token**' -d '{ "attrs": {}, "date": "2017-03-10", "time": "string", "repeat": "string", "days": [0], "start_date": "2017-03-10", "end_date": "2017-03-10", "enabled": true, "remark": "string" }'
‘https://euapi.gizwits.com/app/devices/**{did}**/scheduler/**{id}**'

Device Monitoring

GET /app/devdata/{did}/latest

Get the latest status of the device

⟶ **Header**
'Accept: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'

⟶ **URL**
https://euapi.gizwits.com/app/devdata/**{did}**/latest'

⟶ **Curl Command**
curl -X GET --header 'Accept: application/json' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**' 'https://euapi.gizwits.com/app/devdata/**{did}**/latest'

⟶ **Example returned value :**
{
"did": "string",
"updated_at": "string",
"attr": {}
}

GET /app/datapoint

Get the product data point definition

⟶ **Header**
'Accept: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'

⟶ **URL**
https://euapi.gizwits.com/app/datapoint?product_key=**Product_Key**

⟶ **Curl Command**
curl -X GET --header 'Accept: application/json' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**'
'https://euapi.gizwits.com/app/datapoint?product_key=**Product_Key**'

→ **Example returned value :**


```
{
"name": "string",
"entities": [
{
"id": 0,
"name": "string",
"display_name": "string",
"attrs": [
{
"id": 0,
"name": "string",
"display_name": "string",
"desc": "string",
"type": "W",
"data_type": "bool",
"uint_spec:": {
"min": 0,
"max": 0,
"ratio": 0,
"addition": 0
},
"enum": [
"string"
],
"position": {
"bit_offset": 0,
"len": 0,
"unit": "bit",
"byte_offset": 0
}
}
]
}
],
"protocolType": "string",
"product_key": "string",
"packetVersion": "string",
"ui": {
"object": {
"version": 0,
"showEditButton": true
},
"sections": [
{
```

"elements": [
{
"title": "string",
"key": "string",
"type": "QBooleanElement",
"keyboardType": "NumbersAndPunctuation",
"maxLength": 0,
"items": [
"string"
],
"boolValue": true,
"bind": "string",
"maximumValue": 0,
"minimumValue": 0,
"value": 0,
"object": {
"action": "string",
"bind": [
"string"
],
"perm": "W",
"unit_spec": {
"max": 0,
"step": 0,
"min": 0
}
}
}
]
}
]
}
}

GET /app/devices/{did}

Get device details

→ **Header**
'Accept: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'
'X-Gizwits-User-token: **User_Token**'

→ **URL**
https://euapi.gizwits.com/app/devices/**{did}**

⟶ Curl Command

curl -X GET --header 'Accept: application/json' --header 'X-Gizwits-User-token: **User_Token**' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**' 'https://euapi.gizwits.com/app/devices/**{did}**'

⟶ Example returned value

{
"product_key": "string",
"did": "string",
"mac": "string",
"is_online": true,
"passcode": "string",
"host": "string",
"port": 0,
"port_s": 0,
"ws_port": 0,
"wss_port": 0,
"remark": "string",
"is_disabled": true,
"type": "string",
"dev_alias": "string",
"proto_ver": "01"
}

POST /app/control/{did}

Control the device status

⟶ Header

'Accept: application/json'
'Content-Type: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'
'X-Gizwits-User-token: **User_Token**'

⟶ Heatzy-Pilote Datapoints

{
"raw": [mode]
}

The value for "raw" are :

- "raw": [1,1,0] → Comfort Mode
- "raw": [1,1,1] → ECO mode
- "raw": [1,1,2] → Anti-Freeze mode

- "raw": [1,1,3] → OFF mode

⟶ Heatzy-Pilote second generation datapoints

p1_data1 to p1_data12 : Monday programmation
p2_data1 to p2_data12 : Tuesday programmation
p3_data1 to p3_data12 : Wednesday programmation
p4_data1 to p4_data12 : Thursday programmation
p5_data1 to p5_data12 : Friday programmation
p7_data1 to p6_data12 : Saturday programmation
p7_data1 to p7_data12 : Sunday programmation
mode : 0.cft 1.eco 2.fro 3.stop 4.cft1 5.cft2
derog_mode : derog_mode=1 holiday mode, derog_mode=2 boost mode
derog_time : in minute is derog_mode = 2, in days if derog_mode = 1
lock_switch : 1 --> locked, 2 --> unlocked
time_week : represent day in week (Monday,...)
time_hour : current hour
time_min : current minute
timer_switch : activate / desactivate programmation (0/1)

→ duo Flam / Plugzy datapoints

```
{
"attrs": {
        "on_off": bool
        "mode": "Enum"
        "cft_tempL": number
        "cft_tempH": number
        "eco_tempL": number
        "eco_tempH": number
        "cur_tempL": number
        "cur_tempH": number
}
```

"on_off" represent the product Plugzy. Put 1 to open it, 0 to close.
"mode" represent the Flam. The values are "cft", "eco", "fro" and "stop"
The comfort and eco temperature are coded by two bytes. Each byte represent a number from **0 to 255** where an increment of 1 represent **0.1°C**.
Exemple:
To put the comfort temperature to **26°C**, you need use the value **260** which is coded in binary by **00000001 00000100**. The first byte represent "cft_tempH" which is equal to 1. The second byte represent "cft_tempL" which is equal to **4**.

"cur_tempL" and "cur_tempH" represent the temperature observed by plugzy with the same method explained previously.

→ **INEA Body**

```
{
"attrs": {
        "on_off": bool
        "mode": "Enum"
        "cft_tempL": number
        "cft_tempH": number
        "eco_tempL": number
        "eco_tempH": number
        "cur_tempL": number
        "cur_tempH": number
}
```

⟶ **URL**

https://euapi.gizwits.com/app/control/{did}

⟶ **Curl Command**

curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'X-Gizwits-User-token: **User_Token**' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**' -d '{"raw": [1,1,0]} ' 'https://euapi.gizwits.com/app/control/{**did}**'

User Management

GET /app/users

Get user information

⟶ **Header**

'Accept: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'
'X-Gizwits-User-token: **User_Token**'

⟶ **URL**

https://euapi.gizwits.com/app/users

⟶ Curl Command
curl -X GET --header 'Accept: application/json' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**' --header 'X-Gizwits-User-token: **User_Token**' 'https://euapi.gizwits.com/app/users'

POST /app/users

Create new user

⟶ Header
'Accept: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'

⟶ Body
{
"username": "string",
"password": "string",
"lang": "en",
}

⟶ URL
https://euapi.gizwits.com/app/users

⟶ Curl Command
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**' -d '{"username": "string", "password": "string", "lang": "en",}' 'https://euapi.gizwits.com/app/users'

⟶ Example returned value
{
"uid": "string",
"token": "string",
"expire_at": 0
}

PUT /app/users

Change Password: Enter old_pwd, new_pwd

⟶ Header

'Accept: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'
'X-Gizwits-User-token: **User_Token**'

⟶ **Body**
{
"username": "string",
"password": "string",
"old_pwd": "string",
"new_pwd": "string",
"lang": "en",
}

⟶ **URL**
https://euapi.gizwits.com/app/users

⟶ **Curl Command**
curl -X PUT --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**' --header 'X-Gizwits-User-token: **User_Token**' -d '{ "username": "string", "password": "string", "old_pwd": "string", "new_pwd": "string", "lang": "en"}' 'https://euapi.gizwits.com/app/users'

POST /app/login

User login

⟶ **Header**
'Content-Type: application/json'
'Accept: application/json'
'X-Gizwits-Application-Id: **Heatzy_Application_Id**'

⟶ **Body**
{
"username": "string",
"password": "string",
"lang": "en"
}

⟶ **URL**
https://euapi.gizwits.com/app/login

⟶ **Curl Command**
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**' -d '{ "username": "string", "password": "string", "lang": "en" }' 'https://euapi.gizwits.com/app/login'

⟶ **Example returned value**
{
"uid": "string",
"token": "string",
"expire_at": 0
}

POST /app/reset_password

Send an email or a phone message to reset the user password

⟶ **Header**
'Content-Type: application/json'
'Accept: application/json'

'X-Gizwits-Application-Id: **Heatzy_Application_Id**'

⟶ **Body**
{
"email": "string",
"new_pwd": "string",
}

⟶ **URL**
https://euapi.gizwits.com/app/reset_password

⟶ **Curl Command**
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'X-Gizwits-Application-Id: **Heatzy_Application_Id**' -d '{"email": "string", "new_pwd": "string"}' 'https://euapi.gizwits.com/app/reset_password'