

REST API v1.0

???????????????? REST API

Внимание! Не подключайте контроллер в публичную сеть Интернет! Вместо этого используйте локальную сеть, или выделенную подсеть для безопасного администрирования контроллера и прокси сервера (Nginx, Apache как обратный прокси).

??????? ??????? ???????

Каждый ответ от контроллера формируется в виде JSON строки и имеет вид

```
{
  "success": true,
  "data": {...}
}
```

Свойство data имеет внутри состояние запрашиваемого функционала.

??????? ??????????

Получение системной информации

```
GET /api/systeminfo
```

Ответ

```
{
  "success": true,
  "data": {
    "hostname": "umni-838a74",
    "capabilities": [
      "ethernet",
      "sdcard",
    ]
  }
}
```

```
"webserver",
"mqtt",
"opentherm",
"rf433",
"onewire",
"alarm",
"ntc1",
"ntc2",
"ai1",
"ai2",
"opencollectors",
"oc1",
"oc2",
"buzzer",
"inputs",
"inp1",
"inp2",
"inp3",
"inp4",
"inp5",
"inp6",
"outputs",
"out1",
"out2",
"out3",
"out4",
"out5",
"out6",
"out7",
"out8"
],
"networks": [
  {
    "name": "ethernet",
    "mac": "44:1d:64:83:8a:77",
    "ip": "192.168.88.122",
    "mask": "255.255.252.0",
    "gw": "192.168.88.9",
    "active": true
  }
]
```

```

],
"heap": {
  "total": 380928,
  "free": 190464,
  "min": 166876
}
}
}

```

??????????

hostname	имя устройства	строка
capabilities	список доступных возможностей контроллера	массив строк
networks	список сетевых подключений	объект
heap	список общей и доступной оперативной памяти	объект

????????????? ?????????????? (capabilities)

ethernet	устройство использует проводное подключение
wifi	устройство использует wifi
sdcard	устройство использует microSD карту памяти
webserver	устройство использует веб-сервер
mqtt	устройство использует клиента mqtt
opentherm	устройство использует opentherm (для газовых котлов)
rf433	устройство использует беспроводной канал 433 МГц
onewire	устройство использует onewire
alarm	у устройства есть вывод по датчики наличия 12в
ntc1	у устройства есть NTC термистор №1 (резистор 10K)
ntc2	у устройства есть NTC термистор №2 (резистор 10K)

ai1	у устройства имеется аналоговый вход №1
ai2	у устройства имеется аналоговый вход №2
opencollectors	устройство имеет выходы "открытый коллектор"
oc1	устройство имеет выход открытого коллектора №1
oc2	устройство имеет выход открытого коллектора №2
buzzer	устройство имеет на своем борту бипер (buzzer)
inputs	устройство имеет цифровые входы
inp1	устройство имеет цифровой вход №1
inp2	устройство имеет цифровой вход №2
inp3	устройство имеет цифровой вход №3
inp4	устройство имеет цифровой вход №4
inp5	устройство имеет цифровой вход №5
inp6	устройство имеет цифровой вход №6
outputs	устройство имеет цифровые выходы (релейные выходы)
out1	устройство имеет выход №1
out2	устройство имеет выход №2
out3	устройство имеет выход №3
out4	устройство имеет выход №4
out5	устройство имеет выход №5
out6	устройство имеет выход №6
out7	устройство имеет выход №7
out8	устройство имеет выход №8

?????????? ???????????????

Возвращает JSON объект конфигурации для указанной секции

```
GET /api/conf?section={string}
```

?????? ??????? ?????????????????????????????????????

adc

```
{
  "success": true,
  "data": {
    "channels": [
      {
        "id": 0,
        "label": "ADC Input 1",
        "active": true
      },
      {
        "id": 1,
        "label": "ADC Input 2",
        "active": true
      }
    ]
  }
}
```

ntc

```
{
  "success": true,
  "data": {
    "channels": [
      {
        "id": 0,
        "label": "NTC Sensor 1",
        "active": true
      },
      {
        "id": 1,
        "label": "NTC Sensor 2",
        "active": true
      }
    ]
  }
}
```

```
    }
  ]
}
}
```

dio

```
{
  "success": true,
  "data": {
    "inputs": [
      {
        "config_index": 1,
        "port_index": 1,
        "label": "Input 1 (port 1)",
        "active": true
      },
      ...
    ],
    "outputs": [
      {
        "config_index": 1,
        "port_index": 0,
        "label": "Output 1 (port 0)",
        "active": true,
        "default_state": 0
      },
      ...
    ]
  }
}
```

onewire

```
{
  "success": true,
  "data": {
    "sensors": [
      {
        "sn": "F7062454B2EE5528",
```

```
    "label": "Sensor F7062454B2EE5528",
    "active": true
  }
]
}
}
```

rf433

```
{
  "success": true,
  "data": {
    "devices": []
  }
}
```

???????????? ?????????????? (?????????)

POST /api/settings

Тело запроса

```
{
  "setting": "{setting}",
  "values": {...{values}}
}
```

setting	ключ настройки
values	значения настройки (объект)

???? ???? POST ?????????? ??? ?????? ????????????

mqtt

```
{
  "en": bool,
  "host": string,
  "port": int,
  "user": ?string,
  "password": ?string
}
```

webhook

```
{
  "en": bool,
  "url": string
}
```

outputs

```
{
  "en": bool,
  "label": ?string,
  "index": int
}
```

inputs

```
{
  "en": bool,
  "label": ?string,
  "index": int
}
```

ntc

```
{
  "channel":int,
  "active": bool,
  "offset":float,
  "label":string
}
```

adc

```
{
  "channel":int,
  "active": bool,
  "offset":float,
  "label":string
}
```

onewire

```
{
  "serial": string,
  "label": string,
  "active" : bool,
  "calibration": float
}
```

rf433

```
{
  "mode": ?string, //null or delete
  "serial": number,
  "label": string,
  "alarm": bool,
  "type": number
}
```

opentherm

```
{
  "en": bool,
  "ch_en": bool,
  "ch_sp": int,
  "dhw_en": bool,
  "dhw_sp": int,
  "ch2_en": bool,
  "cool_en": bool,
  "mod": int,
  "otc_en": bool
}
```

Пояснения по параметрам

en

Активен ли Opentherm в контроллере. Вы можете отключить и не использовать этот функционал.

ch_en	Флаг активации режима отопления. При установке значения true, включится режим нагрева теплоносителя на контур отопления.
ch_sp	Пороговое значение температуры теплоносителя, которое будет поддерживаться в контуре отопления при нагреве теплоносителя.
dhw_en	Флаг активации режима горячего водоснабжения. Если отключен, котел не будет включать контур горячей воды или переключать трехходовой клапан (Buderus U072)
dhw_sp	Пороговое значение температуры теплоносителя в контуре горячего водоснабжения.
ch2_en	Второй контур отопления.
cool_en	Охлаждение
mod	Уровень модуляции (0-100)
otc_en	Outside temperature compensation. Режим компенсации температуры по наружному датчику.

Конфигурация устройства (чтение)

```
GET /api/configuration
```

Пример ответа

```
{
  "success": true,
  "data": {
    "title": "My Device",
    "username": "admin",
    "password": null,
    "ntp": "pool.ntp.org",
    "timezone": "MSK-3",
    "socket_port": 8080,
    "token": "secret123",
    "network_mode": 1,
    "wifi_sta_ssid": "HomeWiFi",
    "wifi_sta_password": null,
    "wifi_sta_ip_type": 2,
    "wifi_sta_ip": "192.168.1.100",
    "wifi_sta_netmask": "255.255.255.0",
    "wifi_sta_gateway": "192.168.1.1",
    "wifi_sta_dns": "8.8.8.8",
    "wifi_ap_ssid": "HomeWiFi",
    "wifi_ap_password": null,
    "eth_ip_type": 1,
    "eth_ip": "192.168.1.200",
    "eth_netmask": "255.255.255.0",
```

```
"eth_gateway": "192.168.1.1",
"eth_dns": "8.8.8.8"
}
}
```

???????????????? ???? (?????????????)

POST /api/configuration

Тело запроса (элементы можно передавать не все, а выборочно)

```
{
  // Системные настройки
  "title": "My Device",
  "username": "admin",
  "password": "1234",
  "ntp": "pool.ntp.org",
  "timezone": "MSK-3",
  "socket_port": 514,
  "token": "secret_token",

  // Настройки сети
  "network_mode": 1,          // 1-ETH, 2-WIFI_AP

  // Настройки WiFi STA
  "wifi_sta_ssid": "MyWiFi",
  "wifi_sta_password": "pass123",
  "wifi_sta_ip_type": 1,     // 1-DHCP, 2-STATIC
  "wifi_sta_ip": "192.168.1.100",
  "wifi_sta_netmask": "255.255.255.0",
  "wifi_sta_gateway": "192.168.1.1",
  "wifi_sta_dns": "8.8.8.8",

  // Настройки WiFi AP
  "wifi_ap_ssid": "UMNI_AP",
  "wifi_ap_password": "",
  "wifi_ap_channel": 6,
  "wifi_ap_max_connections": 4,
  "wifi_ap_hidden": false,

  // Настройки Ethernet
```

```
"eth_ip_type": 1,           // 1-DHCP, 2-STATIC
"eth_ip": "192.168.1.100",
"eth_netmask": "255.255.255.0",
"eth_gateway": "192.168.1.1",
"eth_dns": "8.8.8.8"
}
```

Внимание! После установки token, все запросы к API должны идти с заголовком 'Authorization': 'Bearer my_token', либо с передачей параметра запроса &token=my_token в URL.

????????????? ?????????? ??????????

```
POST /api/switch
```

Тело запроса

```
{
  "mode": "{mode}",
  "index": 1,
  "level": true|false
}
```

Возможные значения mode:

- outputs
- opencollectors

Возможные значения index:

- для outputs 1-8

Возможные значения level 0-1

????????????? ???????? Wifi cntnq

```
POST /api/wifi/scan
```

Пример ответа

```
{
  "success": true,
```

```
"data": [  
  {  
    "ssid": string,  
    "rssi": int,  
    "channel": int,  
    "authmode": int,  
    "authmode_name": string  
  }  
]
```

?????? ? ?????????????? (?????????????????)

```
POST /api/beep
```

тело запроса

```
{  
  "count": int, // >= 16  
  "on_ms":int // 0 >= 1000,  
  "off_ms":int // 0 >= 1000  
}
```

?????? ?????? ??????????

```
POST /api/state
```

Тело запроса

```
{  
  "capability": string  
}
```

Формат ответа

```
{  
  "success": true,  
  "data": {  
    "state": {  
      "value": 0  
    }  
  }  
}
```

```
},
"history": {
  "name": string,
  "timestamps": [ ],
  "values": [ ],
  "count": int
}
}
}
```

??????????????

Контроллеры поддерживают автоматизации - это правила в виде JSON данных, которые выполняются при срабатывании определенных условий (состоянии сенсоров).

????????? ?????? ????????????????

GET /api/automations

Ответ

```
[
  {
    "id": 1,
    "if": {
      "capability": "ntc1",
      "op": ">",
      "value": 30.0
    },
    "then": [
      { "capability": "out1", "action": 1 }
    ],
    "else": [
      { "capability": "out1", "action": 0 }
    ]
  }
]
```

????????? ????????????????

POST /api/automations

Тело запроса

```
{
  "if": {
    "capability": "ntcl",
    "op": ">",
    "value": 30.0
  },
  "then": [
    { "capability": "out1", "action": 1 }
  ],
  "else": [
    { "capability": "out1", "action": 0 }
  ]
}
```

?????????? ????????????????

```
PUT /api/automations/{id}
```

Тело запроса

```
{
  "if": {
    "capability": "ntcl",
    "op": ">",
    "value": 35.0
  },
  "then": [
    { "capability": "out1", "action": 1 },
    { "capability": "opentherm", "subtype": "ch", "action": 1 }
  ],
  "else": [
    { "capability": "out1", "action": 0 }
  ]
}
```

?????????? ????????????????

```
DELETE /api/automations/{id}
```

?????????? ?????????? - ?????????? capabilities

opentherm

```
{
  "success": true,
  "state": {
    "ot_enabled": true,
    "ready": true,
    "adapter_success": false,
    "status_code": 3,
    "ch_enable": false,
    "ch_setpoint_requested": 60,
    "dhw_enable": true,
    "dhw_setpoint_requested": 50,
    "otc_enable": false,
    "cooling_enable": false,
    "ch2_enable": false,
    "modulation_level_set": 0,
    "heat_curve_ratio": 0,
    "central_heating_active": false,
    "hot_water_active": false,
    "flame_on": false,
    "is_fault": false,
    "boiler_temperature": 0,
    "return_temperature": 0,
    "dhw_temperature": 0,
    "outside_temperature": 0,
    "dhw_setpoint_current": 0,
    "ch_max_setpoint": 0,
    "modulation": 0,
    "pressure": 0,
    "flow_rate": 0,
    "flow_rate_ch2": 0,
    "fault_code": 0,
    "boiler_config": {
      "control_type": "ON/OFF",
      "dhw_present": false,
      "dhw_config": "INSTANTANEOUS",
      "ch2_present": false,
      "cooling_supported": false,
      "pump_control_allowed": false,
      "slave_ot_version": 0,
      "slave_product_version": 0
    },
    "bounds": {
      "ch": {
```

ntc1	<pre>{ "success": true, "data": { "state": { "value": 27.62923812866211 }, "history": { "name": "ntc1", "timestamps": [1777483230583], "values": [27.62923812866211], "count": 1 } } }</pre>
ntc2	аналогично ntc1
ai1	аналогично ntc1
ai2	аналогично ntc1

Revision #28

Created 2026-03-06 14:05:25 UTC by Михаил Сазанов

Updated 2026-06-02 16:14:51 UTC by Михаил Сазанов