

WireGeo MQTT API

Любое внешнее устройство может обмениваться данными с облаком WireGeo по протоколу MQTT. Для этого в дереве объектов WireGeo необходимо добавить контроллер «MQTT-брокер». Все папки и переменные, созданные внутри данного устройства, будут автоматически транслированы в подключенные MQTT-клиенты. После создания контроллера «MQTT-брокер» необходимо задать его настройки:

Устройство: MQTTBridge2

Полное имя: MQTTBridge2

ID: 630975

Принадлежность: Аккаунт

Имя объекта: MQTTBridge2

ID устройства: MQTTBridge2

Интервал обновления: 0 сек.

Тип контроллера: MQTT-брокер

Параметры MQTT

Адрес брокера: 178.57.222.149 (порт 1883)

Логин: MQTTBridge @test

Пароль:

Корневой топик: test/MQTTBridge2/

Кодировка: UTF-8

Автоматически создавать несуществующие топики

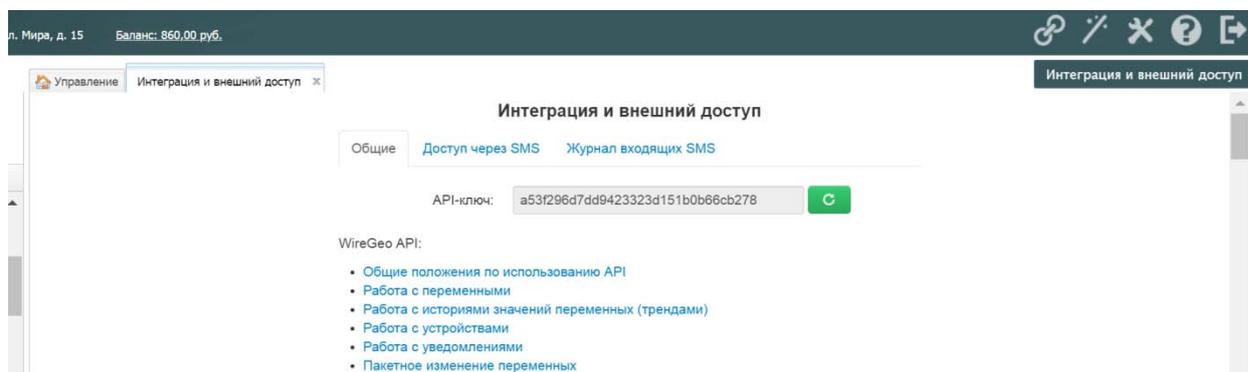
Сохранить Отмена

Если установлена галочка «Автоматически создавать несуществующие топики», то все новые топики, опубликованные подключенными MQTT-клиентами, будут автоматически преобразованы в папки и переменные облака WireGeo.

WireGeo RESTful API

Общие положения по использованию API

Для подключения к облаку WireGeo по протоколу HTTP RESTful необходимо получить API-ключ аккаунта. Для этого кликните на иконку «Интеграция и внешний доступ» в верхнем правом углу Дизайнера WireGeo и скопируйте ключ из поля API-ключ. При необходимости можно сгенерировать новый ключ, нажав на зеленую кнопку рядом с полем ключа. Используйте этот ключ в соответствующем поле HTTP-запросов.



В WireGeo API используется архитектурный стиль REST-взаимодействия для работы с данными и объектами системы (для общего ознакомления с данным подходом рекомендуем прочитать [статью](#) и материалы [сайта](#)). Поддерживаются следующие методы работы (если в соответствующем разделе не указано иное):

- **GET** — получить параметры и состояние объекта;
- **POST** — создать новый объект;
- **PUT** — изменить параметры объекта;
- **DELETE** — удалить объект.

Обязательным входным параметром для всех запросов является **token** — уникальный ключ, присвоенный аккаунту (см. выше). Форматом входных и выходных данных принят JSON. Выходные данные могут быть также отданы в форматах XML и CSV — для этого нужно указать желаемый формат в параметре **format**.

Корневой адрес для API запросов: <https://cloud.wiregeo.com/api/v1/> Обращение к конкретным объектам и данным осуществляется с добавлением к корневому адресу дополнительного пути (см. примеры). Использование HTTPS при работе с API является предпочтительным, но не обязательным. Корневой адрес с работой без защищённого протокола: <http://cloud.wiregeo.com/api/v1/>

Кодировка входных и выходных данных: UTF-8.

Для упрощения работы, а также в целях отладки при выполнении запросов параметр **token** может быть передан не только в теле запроса, но и просто как GET-параметр (т.е. в URL). Метод работы аналогично может быть передан не в HTTP-заголовке, а как GET-параметр **method**.

Пример: <https://cloud.wiregeo.com/api/v1/variables?method=post&token=...> (остальные параметры — в теле запроса, как это описано далее).

Примечание: при выполнении GET-запросов без указания ID объекта выдаются все объекты данного типа из аккаунта. При указании ID или имени выдаётся только один указанный объект. Формат обращения задаётся в URL — либо ByID, либо ByName (см. примеры).

Работа с переменными

Адрес для запросов: <https://cloud.wiregeo.com/api/v1/variables>

Параметры объекта «Переменная»:

- **id** — ID переменной (можно увидеть в окне свойств переменной);
- **name** — имя переменной;
- **full_name** — полное имя (путь внутри системы, разделитель — точка);
- **parent_id** — ID родительского объекта (0 — глобальная переменная);
- **parent** — имя родительского объекта переменной (пустая строка — глобальная переменная);
- **type** — тип данных переменной (A — текстовый, B — логический, N — числовой, U — универсальный);
- **value** — значение (для логического значения необходимо использовать: 1 — истина, 0 — ложь);
- **history** — признак необходимости хранить историю значений (1 — да, 0 — нет);
- **delta** — величина отклонения, при которой значение сохраняется в историю;
- **changed** — время последнего изменения переменной (формат UNIX_TIMESTAMP).

Чтение (GET)

Входные параметры: **token** (в теле запроса), **ByID+ID** или **ByName+Имя** (в URL).

Выходные параметры: все перечисленные.

Примеры вызовов:

- <https://cloud.wiregeo.com/api/v1/variables> — выдаст все переменные аккаунта;
- <https://cloud.wiregeo.com/api/v1/variables/ByID/1234> — выдаст переменную с номером 1234;
- <https://cloud.wiregeo.com/api/v1/variables/ByName/Контроллер.Устройство.Канал1> — выдаст переменную с именем *Контроллер.Устройство.Канал1*.

Пример ответа (аккаунт содержит две переменные):

```
[{"id":"1111","name":"Light1","full_name":"Room.Light1","parent_id":"1110","parent":"Room",  
"type":"N","value":"500","history":0,"delta":"","changed":"1396455022"}, {"id":"1112",  
"name":"Light2","full_name":"Room.Light2","parent_id":"1110","parent":"Room","type":"N",  
"value":"400","history":1,"delta":"10","changed":"1392127912"}]
```

Добавление (POST)

Входные параметры: **token**, **name**, **parent** ИЛИ **parent_id**, **type**, **value**, **history**, **delta** (всё в теле запроса).

Выходные параметры: **id** — ID добавленной переменной, **result** (*OK* в случае успешного запроса, *FAIL* в случае ошибки), **error** (описание ошибки).

Пример запроса:

```
{"token":"a53f296d7dd9423323d151b0b66cb278","name":"Sensor1","type":"N","value":"99",  
"history":"1","delta":"5","parent":"Floor1"}
```

Ответ:

```
{"id":"1234","result":"OK","error":""}
```

Пример ответа в случае ошибки (переменная с таким названием уже существует):

```
{"id":"0","result":"FAIL","error":"OBJECT ALREADY EXISTS"}
```

Изменение (PUT)

Входные параметры: **token**, **name**, **parent** ИЛИ **parent_id**, **type**, **value**, **history**, **delta** (в теле запроса), **ByID+ID** или **ByName+Имя** (в URL).

Выходные параметры: **result** (*OK* в случае успешного запроса, *FAIL* в случае ошибки), **error** (описание ошибки).

Примечание: при изменении значения переменной через PUT-запрос значение автоматически будет занесено в историю, если параметры *history* и *delta* настроены соответствующим образом.

Пример запроса:

```
{"token":"a53f296d7dd9423323d151b0b66cb278","name":"Sensor1","type":"N","value":"105",
"history":"1","delta":"10","parent_id":"123"}
```

Ответ:

```
{"result":"OK","error":""}
```

Пример ответа в случае ошибки (некорректное имя переменной):

```
{"result":"FAIL","error":"NOT VALID NAME"}
```

Дополнение: допускается сокращённая форма PUT-запроса для изменения лишь значения переменной: **token**, **value** (в теле запроса), **ByID+ID** или **ByName+Имя** (в URL), например:

```
{"token":"a53f296d7dd9423323d151b0b66cb278","value":"150"}
```

Пакетное изменение переменных (POST)

Адрес для запросов: <https://cloud.wiregeo.com/api/v1/multi>

Входные параметры: **token**, **data**, (**autocreate**).

data — массив, содержащий переменные для изменения. Каждый элемент массива представляет собой массив из двух элементов: первый элемент — полное имя переменной, второй элемент — новое значение.

autocreate — необязательный параметр, позволяющий активировать автоматическое создание переменных с указанными именами в случае их отсутствия (1 — создавать, 0 — не создавать, по умолчанию переменные НЕ создаются). Переменные, создаваемые автоматически, имеют тип данных «Универсальный». Если полный путь к переменной включает несуществующие уровни иерархии, то они создаются в виде папок.

Выходные параметры: **result** (*OK* в случае успешного запроса, *FAIL* в случае ошибки), **count** (количество фактически обработанных переменных).

За один запрос можно изменять значения не более чем 100 переменных.

Пример запроса:

```
{"token":"a53f296d7dd9423323d151b0b66cb278","data":[[{"Var1","111"},{"Var2","222"},{"Var3","333"}]}
```

Ответ:

```
{"result":"OK","count":3}
```

Пример ответа в случае ошибки:

```
{"result":"FAIL","count":0}
```

Удаление (DELETE)

Входные параметры: **token**, (в теле запроса), **ByID+ID** или **ByName+Имя** (в URL).

Выходные параметры: **result** (*OK* в случае успешного удаления, *FAIL* в случае ошибки), **error** (описание ошибки).

Важно: при удалении переменной удаляется вся её история.

Пример ответа:

```
{"result":"ОК","error":""}
```

Работа с историями значений переменных (трендами)

Адрес для запросов: <https://cloud.wiregeo.com/api/v1/history>

Чтение (GET)

Входные

параметры: **token**, **time_begin**, **time_end**, **limit_count**, **group**, **group_function**, **values**, **time_format** (в теле запроса), **ByID+ID** или **ByName+Имя** (в URL).

Описание параметров:

- **time_begin** — начало времени извлекаемого диапазона (в формате DATETIME или UNIX_TIMESTAMP);
- **time_end** — конец времени извлекаемого диапазона (в формате DATETIME или UNIX_TIMESTAMP);
- **limit_count** — максимальное количество выдаваемых значений (если в заданном интервале больше сохранённых значений, то исключаются самые старые), но не больше 10 000;
- **group** — группировка значений по интервалам времени, возможные значения: *none* (без группировки), *day* (группировка по дням) или *month* (группировка по месяцам);
- **group_function** — способ группировки значений: *min* (выбирается минимальное значение из интервала), *max* (максимальное) или *avg* (среднее, используется по умолчанию);
- **values** — выводимые значения: *absolute* — абсолютные, *relative* — относительные (первое является абсолютным, все последующие — изменения относительно предыдущих);
- **time_format** — *unix* или *datetime* — формат вывода времени в ответе (выдаётся DATETIME или UNIX_TIMESTAMP соответственно).

Выходные данные: массив пар вида «время-значение».

Пример запроса:

```
{"token":"a53f296d7dd9423323d151b0b66cb278","time_begin":"2016-01-01 00:00:00",  
"time_end":"2016-01-05 23:59:59","group":"day","group_function":"max","values":"absolute",  
"time_format":"datetime"}
```

Ответ:

```
[["2016-01-01",24.7],["2016-01-02",32.5],["2016-01-03",28.8],["2016-01-04",23.5],["2016-01-05",20.1]]
```

Добавление (POST)

Метод позволяет добавить значения в историю переменной (текущее значение переменной при этом не меняется).

Входные параметры: **token**, **values** (в теле запроса), **ByID+ID** или **ByName+Имя** (в URL).

Массив **values** представляет значения, которые необходимо сохранить в истории. Каждый элемент массива является одной точкой: первый элемент массива — значение переменной, второй — время этого значения в формате UNIX Timestamp.

Примечание: если требуется сохранить только одно значение, оно всё равно должно быть передано как массив из одного элемента.

Пример запроса:

```
{"token":"a53f296d7dd9423323d151b0b66cb270","values":[[10,1455819637],[20,1455819037],[30,1455818637]]}
```

Выходные параметры: **result** (*OK* в случае успешного запроса, *FAIL* в случае ошибки), **error** (описание ошибки), **count** (количество добавленных в историю значений).

Пример ответа:

```
{"result":"OK","error":"","count":3}
```

Работа с устройствами

В терминах WireGeo к устройствам относятся:

1. контроллеры;
2. непосредственно устройства;
3. интерфейсы устройств (например, RS-485, 1-Wire и т.д.).

Параметры объекта «Устройство»:

- **id** — ID устройства (можно увидеть в окне свойств устройства);
- **name** — имя устройства;
- **full_name** — полное имя (путь внутри системы, разделитель — точка);
- **parent_id** — ID родительского объекта (0 — устройство находится в корне аккаунта);
- **parent** — имя родительского объекта переменной (пустая строка — устройство находится в корне аккаунта);
- **type** — тип устройства (1 — контроллер, 2 — устройство, 3 — интерфейс);
- **system** — системное имя устройства;
- **properties** — параметры устройства (массив в формате JSON, например для контроллера может быть `{"controller_type":"wiregeo-smart"}`);
- **dt** — интервал синхронизации (в секундах);
- **active** — признак включенности устройства (1 — включено, 0 — выключено);
- **driver** — имя драйвера.

Чтение (GET)

Входные параметры: **token** (в теле запроса), **ByID+ID** или **ByName+Имя** (в URL).

Выходные параметры: все перечисленные.

Примеры вызовов:

- <https://cloud.wiregeo.com/api/v1/devices> — выдаст все устройства аккаунта;
- <https://cloud.wiregeo.com/api/v1/devices/ByID/1234> — выдаст устройство с номером 1234;
- <https://cloud.wiregeo.com/api/v1/devices/ByName/Контроллер.Устройство> — выдаст устройство с именем *Контроллер.Устройство*.

Пример ответа (одно устройство):

```
[{"id":"1201","name":"Device1","full_name":"CONTROLLER.Device1","parent_id":"1200",
"parent":"CONTROLLER","type":"2","system":"DEVICE-
3458923494","properties":"","dt":"60",
"active":"1","driver":""}]
```

Добавление (POST)

Входные

параметры: **token**, **name**, **parent** ИЛИ **parent_id**, **type**, **system**, **properties**, **dt**, **active**, **driver** (всё в теле запроса).

Выходные параметры: **id** — ID добавленного устройства, **result** (*OK* в случае успешного запроса, *FAIL* в случае ошибки), **error** (описание ошибки).

Пример запроса:

```
{"token":"a53f296d7dd9423323d151b0b66cb278","name":"Устройство1","type":"2",
"driver":"DRV-001","system":"DEVICE-ABC-
999888777","dt":"60","parent":"Room1","active":"1"}
```

Ответ:

```
{"id":"1202","result":"OK","error":""}
```

Пример ответа в случае ошибки (некорректное имя устройства):

```
{"result":"FAIL","error":"NOT VALID NAME"}
```

Изменение (PUT)

Входные

параметры: **token**, **name**, **parent** ИЛИ **parent_id**, **type**, **system**, **properties**, **dt**, **active**, **driver** (в теле запроса), **ByID+ID** или **ByName+Имя** (в URL).

Выходные параметры: **result** (*OK* в случае успешного запроса, *FAIL* в случае ошибки), **error** (описание ошибки).

Пример запроса:

```
{"token":"a53f296d7dd9423323d151b0b66cb278","name":"Устройство1","type":"2",
"driver":"DRV-001","system":"DEVICE-ABC-999888777","dt":"120","parent":"Room1",
"active":"0"}
```

Ответ:

```
{"result":"OK","error":""}
```

Пример ответа в случае ошибки (устройство с указанным именем/ID не существует):

```
{"result":"FAIL","error":"OBJECT DOES NOT EXIST"}
```

Удаление (DELETE)

Входные параметры: **token**, (в теле запроса), **ByID+ID** или **ByName+Имя** (в URL).

Выходные параметры: **result** (*OK* в случае успешного удаления, *FAIL* в случае ошибки), **error** (описание ошибки).

Пример ответа:

```
{"result":"OK","error":""}
```

Работа с уведомлениями

Адрес для запросов: <https://cloud.wiregeo.com/api/v1/notifications>

Параметры объекта «Уведомление»:

- **id** — ID уведомления (можно увидеть в окне свойств уведомления);
- **name** — имя уведомления;
- **email** — адрес для отправки e-mail (можно указывать несколько через запятую);
- **sms** — номер для отправки SMS (можно указывать несколько через запятую);
- **push** — имя приложения для отправки Push-уведомления («*» — все приложения аккаунта);
- **text** — текст уведомления;
- **active** — признак включенности уведомления (1 — включено, 0 — выключено).

Чтение (GET)

Входные параметры: **token** (в теле запроса), **ByID+ID** или **ByName+Имя** (в URL).

Выходные параметры: все перечисленные.

Примеры вызовов:

- <https://cloud.wiregeo.com/api/v1/notifications> — выдаст все уведомления аккаунта;
- <https://cloud.wiregeo.com/api/v1/notifications/ByID/6789> — выдаст уведомление с номером 6789;
- <https://cloud.wiregeo.com/api/v1/notifications/ByName/Notify1> — выдаст уведомление с именем *Notify1*.

Пример ответа:

```
[{"id":"1000","name":"AlarmNotify","email":"ivan@maildomain.ru","sms":"","push":"","text":"Alarm!","active":"1"}]
```

Добавление (POST)

Входные параметры: **token**, **name**, **email**, **sms**, **push**, **text**, **active** (всё в теле запроса).

Выходные параметры: **id** — ID добавленного уведомления, **result** (*OK* в случае успешного запроса, *FAIL* в случае ошибки), **error** (описание ошибки).

Пример запроса:

```
{"token":"a53f296d7dd9423323d151b0b66cb278","name":"NewNotify","email":"ivan@mail-server.ru","sms":"","push":"*","text":"WireGeo online!","active":"1"}
```

Ответ:

```
{"id":"1000","result":"OK","error":""}
```

Пример ответа в случае ошибки (не удалось создать уведомление):

```
{"result":"FAIL","error":"COULD NOT CREATE NOTIFICATION"}
```

Изменение (PUT)

Входные параметры: **token**, **name**, **email**, **sms**, **push**, **text**, **active** (в теле запроса), **ByID+ID** или **ByName+Имя** (в URL).

Выходные параметры: **result** (*OK* в случае успешного запроса, *FAIL* в случае ошибки), **error** (описание ошибки).

Пример запроса:

```
{"token":"a53f296d7dd9423323d151b0b66cb278","name":"NewNotify",  
"email":"anna@mail-server.ru","sms":"","push":"*","text":"WireGeo online!","active":"1"}
```

Ответ:

```
{"result":"OK","error":""}
```

Пример ответа в случае ошибки (не удалось изменить параметры уведомления):

```
{"result":"FAIL","error":"COULD NOT UPDATE NOTIFICATION"}
```

Удаление (DELETE)

Входные параметры: **token**, (в теле запроса), **ByID+ID** или **ByName+Имя** (в URL).

Выходные параметры: **result** (*OK* в случае успешного удаления, *FAIL* в случае ошибки), **error** (описание ошибки).

Пример ответа:

```
{"result":"OK","error":""}
```