



ENERGY & CLIMATE CONTROL

## Описание REST-API PDU2/3/4

---

Версии ПО: 1.16.1 / 3.6.1 / 4.6.1

**Москва - 2025**

## Оглавление

1.	<b>ВВЕДЕНИЕ</b> .....	2
2.	<b>МОНИТОРИНГ И НАСТРОЙКА ЧЕРЕЗ REST-API</b> .....	2
2.1.	АУТЕНТИФИКАЦИЯ .....	2
2.2.	ЗАПРОСЫ ДЛЯ МОНИТОРИНГА ПАРАМЕТРОВ ПИТАНИЯ .....	4
2.3.	ЗАПРОСЫ ДЛЯ МОНИТОРИНГА ДАТЧИКОВ RS-НТ1 .....	8
2.4.	ЗАПРОСЫ ДЛЯ МОНИТОРИНГА УСТРОЙСТВ R-DH1 (DOORHUB).....	10
2.5.	ЗАПРОСЫ ДЛЯ МОНИТОРИНГА REM-ЗАМКОВ.....	13
2.6.	ЗАПРОСЫ ДЛЯ ПОЛУЧЕНИЯ ЖУРНАЛА СОБЫТИЙ .....	15
2.7.	ЗАПРОСЫ ДЛЯ МОНИТОРИНГА СОСТОЯНИЯ СКУД.....	17
2.8.	ЗАПРОСЫ ДЛЯ НАСТРОЙКИ СКУД.....	22

## 1. Введение

Этот документ содержит описание REST-API для контроллеров интеллектуальных блоков распределения питания REM™ второго и последующих поколений (далее – PDU). В контексте описания функций основного контроллера PDU также будет использоваться термин «Контроллер».

## 2. Мониторинг и настройка через REST-API

REST-API позволяет автоматизировать управление и мониторинг СКУД, параметров питания, датчиков температуры и влажности, а также дает возможность интегрировать Контроллер в различные системы верхнего уровня. Контроллер поддерживает протоколы **HTTP**, **HTTPS** и **mTLS** и может быть доступен по разным IP-адресам в зависимости от настроек. Все дальнейшие примеры запросов приведены для протокола http.

Формат URL REST-API запроса: <Протокол>://<IP-адрес>/<Endpoint>

Например: http://192.168.0.254/acs\_zones.fcgi

URL может содержать query параметры, например:

http://192.168.0.254/acs\_door\_delete.fcgi?name=Front%20door

Запрос должен содержать cookie с именем jwt, либо заголовок "Authorization: Bearer" с токеном аутентификации (см 2.1)

### 2.1. Аутентификация

#### Запросы для аутентификации

##### auth.fcgi:

Аутентификация осуществляется путем отправки POST-запроса на URL <Протокол>://<IP-адрес>/auth.fcgi

В теле запроса должна содержаться учетные данные (логин и пароль) в формате ключ=значение. Пример тела запроса: login=admin&password=12345

Пример полного запроса с использованием утилиты curl:

```
curl -X POST http://192.168.0.254/auth.fcgi -d "login=admin&password=12345"
```

##### Ответ сервера на запрос auth.fcgi

При успешной аутентификации сервер вернет ответ с HTTP-статусом 200 и токеном в теле запроса:

```
{
  "token": " eyJh...dQ"
}
```

**signin.fcgi:**

Аутентификация также может осуществляться путем отправки POST-запроса на URL <Протокол>://<IP-адрес>/signin.fcgi

В теле запроса должна содержаться учетные данные (логин и пароль) в формате ключ=значение. Пример тела запроса: login=admin&password=12345

Пример полного запроса с использованием утилиты curl:

```
curl -X POST http://192.168.0.254/signin.fcgi -d "login=admin&password=12345" -D -
```

**Ответ сервера на запрос signin.fcgi**

При успешной аутентификации сервер вернет ответ с HTTP-статусом 303 See Other и установит токен аутентификации в cookie с именем jwt. Значение данного токена используется для последующей идентификации пользователя.

**Использование токена**

Для выполнения остальных запросов к контроллеру, токен, полученный при аутентификации, должен быть передан в заголовке Authorization: Bearer <jwt>, либо в cookie запроса под именем jwt. Без предоставления действующего токена сервер отклонит запрос.

Пример передачи токена с запросом в заголовке Authorization: Bearer:

```
curl -X GET -s -H "Authorization: Bearer eyJh...dQ" "http://192.168.0.254/power_mon.fcgi"
```

Пример передачи токена с запросом в cookie:

```
curl -X GET http://192.168.0.254/acs_zones.fcgi -b "jwt=eyJh...dQ"
```

Токен истекает через настраиваемый таймаут после выдачи. Для продолжения работы потребуется повторная аутентификация. По умолчанию таймаут равен 5 минутам. Изменить таймаут можно через WEB интерфейс на странице Протоколы->HTTP/HTTPS, либо через CLI командой http timeout <minutes>.

## 2.2. Запросы для мониторинга параметров питания

### Единицы измерения:

- Мощность: ватты (Вт)
- Ток: амперы × 100 (A × 100)
- Напряжение: вольты (В)
- Энергия: ватт-часы (Вт·ч)
- Время: Unix-эпохи (количество секунд, прошедших с 00:00:00 UTC 1 января 1970 года)

### Статусы:

- 0 - NORMAL (нормальное состояние)
- 1 - ALMIN (предупреждение)
- 2 - ALMAJ (тревога)

### 2.2.1. Мониторинг питания на контуре

**Описание:** получает данные по параметрам питания на контуре

**Endpoint:** /power\_mon.fcgi

**Тип:** GET

**Query параметры:**

- pdu (обязательный) - индекс PDU (начиная с 0)
- bank (обязательный) - индекс контура (начиная с 0)

**Формат ответа в случае успешного выполнения:**

```
{
  "name": "L1.1",
  "status": 1,
  "breaker_state": 1,
  "voltage": 230,
  "current": 1500,
  "active_power": 3500,
  "active_energy": 25000,
  "timestamp": 1761145827
}
```

**Описание параметров в ответе на запрос:**

- name – Имя контура
- status – Статус контура (0-норма, 1-предупреждение, 2-тревога)
- breaker\_state - Состояние выключающего автомата (0-неизвестно, 1-замкнут, 2-разомкнут). Поле отсутствует, если выключающий автомат не установлен
- voltage – Напряжение на контуре в вольтах
- current - Ток на контуре в амперах × 100
- active\_power – Активная мощность на контуре в ваттах
- active\_energy – Активная энергия на контуре в ватт-часах
- timestamp - Время последнего успешного измерения, в Unix-эпохах

Примечание: при неисправности входного измерителя, в ответе на запрос будут отсутствовать поля voltage, current, active\_power, active\_energy, статус контура будет равен 1 (ALMIN). Поле active\_energy будет отсутствовать, если энергия на контуре не измеряется.

**Пример запроса:**

```
curl -X GET -H "Authorization: Bearer jwt" "http://192.168.0.254/power_mon.fcgi?pdu=0&bank=0"
```

### 2.2.2. Мониторинг питания на фазе

**Описание:** получает данные по параметрам питания на фазе

**Endpoint:** /power\_mon.fcgi

Тип: GET

Query параметры:

- pdu (обязательный) - индекс PDU (начиная с 0)
- phase (обязательный) - индекс фазы (начиная с 0)

Формат ответа в случае успешного выполнения:

```
{
  "name": "L1",
  "voltage": 230,
  "current": 1500,
  "active_power": 3500,
  "active_energy": 46500,
  "timestamp": 1761145827
}
```

Описание параметров в ответе на запрос:

- name – Имя фазы
- status – Статус фазы (0-норма, 1-предупреждение, 2-тревога)
- voltage – Напряжение на фазк в вольтах
- current - Ток на фазе в амперах × 100
- active\_power – Активная мощность на фазе в ваттах
- active\_energy – Активная энергия на фазе в ватт-часах
- timestamp - Время последнего успешного измерения, в Unix-эпохах

Примечание: при неисправности входного измерителя, в ответе на запрос будут отсутствовать поля voltage, current, active\_power, active\_energy, статус фазы будет равен 1 (ALMIN). Поле active\_energy будет отсутствовать, если энергия на фазе не измеряется.

Пример запроса:

```
curl -X GET -H "Authorization: Bearer jwt" "http://192.168.0.254/power_mon.fcgi?pdu=0&phase=0"
```

### 2.2.3. Мониторинг питания на розетке

Описание: получает данные по параметрам питания на розетке

Endpoint: /power\_mon.fcgi

Тип: GET

Query параметры:

- pdu (обязательный) - индекс PDU (начиная с 0)
- load (обязательный) - индекс фазы (начиная с 0)

Формат ответа в случае успешного выполнения:

```
{
  "n": "Socket 2",
  "phc": "L1.2",
  "status": 0,
  "is_gone": false,
  "is_on": false,
  "power": 0,
  "current": 0,
  "energy": 1500,
  "timestamp": 1761145826
}
```

Описание параметров в ответе на запрос:

- n – Имя розетки
- phc - Имя контура, которому принадлежит розетка
- status – Статус розетки (0-норма, 1-предупреждение, 2-тревога)
- is\_gone – Связь с розеткой потеряна (true) или есть (false)

- is\_on - Розетка включена (true) или выключена (false)
- current - Ток на фазе в амперах × 100
- power – Активная мощность на фазе в ваттах
- active\_energy – Активная энергия на фазе в ватт-часах
- timestamp - Время последнего успешного измерения, в Unix-эпохах

Примечание: при неисправности выходного измерителя, в ответе на запрос будут отсутствовать поля voltage, current, active\_power, active\_energy, статус розетки будет равен 2 (ALMAJ), поле is\_gone будет равно true. Поля current, power, active\_energy также будут отсутствовать у розеток без измерения (только с управлением).

#### Пример запроса:

```
curl -X GET -H "Authorization: Bearer jwt" "http://192.168.0.254/power_mon.fcgi?pdu=0&load=1"
```

### 2.2.4. Мониторинг питания на всем PDU

**Описание:** получает данные по параметрам питания на всех контурах, фазах и розетках PDU

**Endpoint:** /power\_mon.fcgi

**Тип:** GET

**Query параметры:**

- pdu (обязательный) - индекс PDU (начиная с 0)

**Формат ответа в случае успешного выполнения:**

```
{
  "n": "PDU-1",
  "total_active_power": 15000,
  "total_active_energy": 125000,
  "banks": [
    {
      "name": "L1.1",
      "status": 2,
      "voltage": 230,
      "current": 1500,
      "active_power": 3500,
      "active_energy": 45000,
      "timestamp": 1761145827
    },
    {
      "name": "L1.2",
      "status": 0,
      "voltage": 230,
      "current": 0,
      "active_power": 0,
      "active_energy": 1500,
      "timestamp": 1761145827
    }
  ],
  "phases": [
    {
      "name": "L1",
      "voltage": 230,
      "current": 1500,
      "active_power": 3500,
      "active_energy": 46500,
      "timestamp": 1761145827
    }
  ],
  "loads": [
    {
```

```
    "n": "Socket 1",
    "phc": "L1.1",
    "status": 2,
    "is_gone": false,
    "is_on": true,
    "power": 3500,
    "current": 1500,
    "energy": 45000,
    "timestamp": 1761145826
  },
  {
    "n": "Socket 2",
    "phc": "L1.2",
    "status": 0,
    "is_gone": false,
    "is_on": false,
    "power": 0,
    "current": 0,
    "energy": 1500,
    "timestamp": 1761145826
  }
]
```

**Пример запроса:**

```
curl -X GET -H "Authorization: Bearer jwt" "http://192.168.0.254/power_mon.fcgi?pdu=0"
```

## 2.3. Запросы для мониторинга датчиков RS-HT1

### Единицы измерения:

- Температура: градусы Цельсия × 10 (например, 235 = 23.5°C)
- Влажность: проценты (0-100)
- Статус: числовое значение (0, 1, 2)
- Время: Unix-эпохи (количество секунд, прошедших с 00:00:00 UTC 1 января 1970 года)

### 2.3.1. Мониторинг одного датчика RS-HT1

**Описание:** возвращает данные одного датчика RS-HT1 по индексу дискретного входа, либо по имени

**Endpoint:** /rsht1\_mon.fcgi

**Тип:** GET

**Query параметры:**

- dinIdx - индекс дискретного входа контроллера
- name – имя датчика RS-HT1

Примечание: передается либо параметр dinIdx, либо параметр name

**Формат ответа в случае успешного выполнения:**

```
{
  "dinIdx": 0,
  "name": "Server Room Sensor 1",
  "status": 0,
  "temperature": 235,
  "humidity": 45,
  "lost": false,
  "dataTimestamp": 1761145827
}
```

**Описание параметров в ответе на запрос:**

- dinIdx – Индекс дискретного входа контроллера
- name – Имя датчика
- status – Статус датчика (0-норма, 1-предупреждение, 2-тревога)
- temperature - Температура в градусах Цельсия × 10
- humidity – Влажность в процентах (0-100)
- lost – Связь с устройством потеряна (true) или есть (false)
- dataTimestamp - Время последнего успешного измерения, в Unix-эпохах

**Примеры запроса:**

```
curl -X GET -H "Authorization: Bearer jwt" "http://192.168.0.254/rsht1_mon.fcgi?dinIdx=0"
```

```
curl -X GET -H "Authorization: Bearer jwt"
"http://192.168.0.254/rsht1_mon.fcgi?name=Server%20Room%20Sensor%201"
```

### 2.3.2. Мониторинг всех включенных датчиков RS-HT1

**Описание:** возвращает данные всех датчиков RS-HT1

**Endpoint:** /rsht1\_mon.fcgi

**Тип:** GET

**Query параметры:** отсутствуют

**Формат ответа в случае успешного выполнения:**

```
[
  {
    "dinIdx": 0,
    "name": "Server Room Sensor 1",
    "status": 0,
    "temperature": 235,
    "humidity": 45,
    "lost": false,
    "dataTimestamp": 1761145827
  },
  {
    "dinIdx": 1,
    "name": "Server Room Sensor 2",
    "status": 1,
    "temperature": 285,
    "humidity": 52,
    "lost": false,
    "dataTimestamp": 1761145827
  },
  {
    "dinIdx": 2,
    "name": "Server Room Sensor 3",
    "status": 2,
    "temperature": 0,
    "humidity": 0,
    "lost": true,
    "dataTimestamp": 1761145820
  }
]
```

**Описание параметров в ответе на запрос:** см 2.3.1

**Пример запроса:**

```
curl -X GET -H "Authorization: Bearer jwt" "http://192.168.0.254/rsht1_mon.fcgi"
```

## 2.4. Запросы для мониторинга устройств R-DH1 (Doorhub)

### Единицы измерения:

- Температура: градусы Цельсия × 10 (например, 235 = 23.5°C)
- Влажность: проценты (0-100)
- Статус: числовое значение (0, 1, 2)
- Время: Unix-эпохи (количество секунд, прошедших с 00:00:00 UTC 1 января 1970 года)

### 2.4.1. Мониторинг одного устройства R-DH1

**Описание:** возвращает данные одного R-DH1 (Doorhub) по индексу внешнего устройства, либо по имени

**Endpoint:** /doorhub\_mon.fcgi

**Тип:** GET

#### Query параметры:

- devIdx - индекс внешнего устройства R-DH1 в настройках контроллера
- name – имя устройства

Примечание: передается либо параметр devIdx, либо параметр name

#### Формат ответа в случае успешного выполнения:

```
[
  {
    "devIdx": 0,
    "name": "DH1",
    "status": 0,
    "lost": false,
    "dataTimestamp": 1715769605,
    "door": {
      "name": "Main Door",
      "open": false
    },
    "sensors": [
      {
        "name": "DH.HT1",
        "template": "RS-HT1",
        "lost": false,
        "t": 242,
        "h": 45,
        "status": 0
      },
      {
        "name": "DH.HT2",
        "template": "RS-HT1",
        "lost": false,
        "t": 244,
        "h": 43,
        "status": 0
      },
      {
        "name": "DH.T3",
        "template": "RS-T1",
        "lost": false,
        "t": 231,
        "status": 0
      }
    ]
  }
]
```

```

    ]
  }
]

```

#### Описание параметров в ответе на запрос:

- devIdx – Индекс внешнего устройства R-DH1 в настройках контроллера
- name – Имя устройства
- status - Статус устройства (0-норма, 1-предупреждение, 2-тревога)
- lost – Связь с устройством потеряна (true) или есть (false)
- dataTimestamp - Время последнего успешного обмена данными с устройством, в Unix-эпохах
- door – Объект с данными о состоянии датчика двери. Поле отсутствует, если дверь отключена
- sensors – Массив включенных датчиков Sensor1 ... Sensor 3

#### Описание полей объекта door (состояние датчика двери):

- name – Имя двери
- open – Дверь открыта (true) или закрыта (false)

#### Описание полей объекта массива sensors (датчиков RS-HT1 и RS-T1)

- name – Имя датчика
- template – тип датчика (RS-HT1 или RS-T1)
- lost – Связь с датчиков потеряна (true) или нет (false)
- t – Температура в градусах Цельсия × 10
- h – Влажность в процентах. Поле присутствует только у датчика RS-HT1
- status - Статус датчика (0-норма, 1-предупреждение, 2-тревога)

#### Примеры запроса:

```
curl -X GET -H "Authorization: Bearer jwt" "http://192.168.0.254/doorhub_mon.fcgi?devIdx=0"
```

```
curl -X GET -H "Authorization: Bearer jwt" "http://192.168.0.254/doorhub_mon.fcgi?name=DH1"
```

### 2.4.2. Мониторинг всех включенных R-DH1

**Описание:** возвращает данные всех включенных в настройках устройств R-DH1

**Endpoint:** /doorhub\_mon.fcgi

**Тип:** GET

**Query параметры:** отсутствуют

**Формат ответа в случае успешного выполнения:**

```

[
  {
    "devIdx": 4,
    "name": "DH1",
    "status": 0,
    "lost": false,
    "dataTimestamp": 1770644191,
    "door": {
      "name": "DH1.Door",
      "open": false
    },
    "sensors": []
  },
  {
    "devIdx": 5,
    "name": "DH2",
    "status": 0,
    "lost": false,
    "dataTimestamp": 1770644191,

```

```
"door": {
  "name": "DH2.Door",
  "open": false
},
"sensors": [
  {
    "name": "DH.HT1",
    "template": "RS-HT1",
    "lost": false,
    "t": 242,
    "h": 45,
    "status": 0
  },
  {
    "name": "DH.T3",
    "template": "RS-T1",
    "lost": false,
    "t": 231,
    "status": 0
  }
]
}
```

**Описание параметров в ответе на запрос:** см 2.4.1

**Пример запроса:**

```
curl -X GET -H "Authorization: Bearer jwt" "http://192.168.0.254/doorhub_mon.fcgi"
```

## 2.5. Запросы для мониторинга REM-замков

### 2.5.1. Мониторинг одного REM-замка

**Описание:** возвращает данные одного REM-замка по индексу внешнего устройства, либо по имени

**Endpoint:** /remlock\_mon.fcgi

**Тип:** GET

**Query параметры:**

- devIdx - индекс внешнего устройства REM-замка в настройках контроллера
- name – имя REM-замка

Примечание: передается либо параметр devIdx, либо параметр name

**Формат ответа в случае успешного выполнения:**

```
{
  "devIdx": 0,
  "name": "Main Entrance Lock",
  "blocked": false,
  "opened": false,
  "lost": false,
  "lastSwipedCard": {
    "id": 123456789,
    "cardType": "mifare_classic",
    "success": true,
    "timestamp": 1715769600
  },
  "lastUnlockTimestamp": 1715769600,
  "dataTimestamp": 1715769605
}
```

**Описание параметров в ответе на запрос:**

- devIdx – Индекс внешнего устройства REM-замка в настройках контроллера
- name – Имя REM-замка
- blocked - Замок заблокирован (true) или разблокирован (false)
- opened – Ручка замка открыта (true) или закрыта (false)
- lost – Связь с устройством потеряна (true) или есть (false)
- lastSwipedCard – Объект с информацией о последней приложенной карте
- id – ID последней приложенной карты
- cardType – Тип последней приложенной карты (mifare\_classic, mifare\_plus, em\_marine, hid\_prox)
- success – Успешный (true) проход по последней приложенной карте или нет (false)
- timestamp – Время последнего прикладывания карты, в Unix-эпохах
- lastUnlockTimestamp – Время последней разблокировки REM-замка в Unix-эпохах
- dataTimestamp - Время последнего успешного опроса REM-замка в Unix-эпохах

**Примеры запроса:**

```
curl -X GET -H "Authorization: Bearer jwt" "http://192.168.0.254/remlock_mon.fcgi?devIdx=0"
```

```
curl -X GET -H "Authorization: Bearer jwt" "http://192.168.0.254/remlock_mon.fcgi?name=REMlock1"
```

### 2.5.2. Мониторинг всех включенных REM-замков

**Описание:** возвращает данные всех включенных в настройках REM-замков

**Endpoint:** /remlock\_mon.fcgi

**Тип:** GET

**Query параметры:** отсутствуют

**Формат ответа в случае успешного выполнения:**

```
[
  {
    "devIdx": 3,
    "name": "AA39LockH",
    "blocked": true,
    "opened": false,
    "lost": false,
    "lastSwipedCard": {
      "id": 494292636,
      "cardType": "mifare_classic",
      "success": true,
      "timestamp": 1763549557
    },
    "lastUnlockTimestamp": 1763549641,
    "dataTimestamp": 1763563902
  },
  {
    "devIdx": 4,
    "name": "AB39LockH",
    "blocked": true,
    "opened": false,
    "lost": false,
    "lastSwipedCard": {
      "id": 494292636,
      "cardType": "mifare_classic",
      "success": true,
      "timestamp": 1763549062
    },
    "lastUnlockTimestamp": 1763549641,
    "dataTimestamp": 1763563902
  },
  {
    "devIdx": 5,
    "name": "AC39LockH",
    "blocked": true,
    "opened": false,
    "lost": false,
    "lastSwipedCard": {
      "id": 764031651,
      "cardType": "mifare_classic",
      "success": false,
      "timestamp": 1763563893
    },
    "lastUnlockTimestamp": 1763549641,
    "dataTimestamp": 1763563903
  },
  {
    "devIdx": 6,
    "name": "AD39LockH",
    "blocked": true,
    "opened": false,
    "lost": false,
    "lastSwipedCard": {
      "id": 3925920955,
      "cardType": "mifare_classic",
      "success": false,
      "timestamp": 1763549219
    },
    "lastUnlockTimestamp": 1763549641,
    "dataTimestamp": 1763563903
  }
]
```

```
}
]
```

**Описание параметров в ответе на запрос:** см 2.5.1

**Пример запроса:**

```
curl -X GET -H "Authorization: Bearer jwt" "http://192.168.0.254/remlock_mon.fcgi"
```

## 2.6. Запросы для получения журнала событий

### 2.6.1. Получить журнал событий

**Описание:** получает журнал событий. Максимальный объем журнала, который может быть получен через этот запрос – 1 мегабайт. Если объем журнала превышает 1 мегабайт, журнал начинает формироваться заново, старый журнал архивируется. Полный журнал (вместе с архивными журналами) можно получить через запрос, описанный в пункте 2.6.2

**Endpoint:** /logs.fcgi

**Тип:** GET

**Код ответа:**

- 200 OK — запрос выполнен успешно.

**Формат ответа в случае успешного выполнения:**

```
2025-11-21 00:43:00 PDU daemon.info [INPUTS]: (F) din1:NORMAL -> ALMAJ,Connection Lost
2025-11-21 00:43:00 PDU daemon.info [INPUTS]: (F) din2:NORMAL -> ALMAJ,Connection Lost
2025-11-21 00:43:00 PDU daemon.info GUARD: (F) 24 hours group sensor has triggered
2025-11-21 00:43:00 PDU daemon.info GUARD: Received new alarm from inputs service = ALMAJ
2025-11-21 00:43:00 PDU daemon.info GUARD: (F) PDU: state changed ALMIN -> ALMAJ
2025-11-21 00:43:00 PDU daemon.info GUARD: Received new alarm from extdev service = NORMAL
2025-11-21 00:43:13 PDU daemon.info [EXTDEV]: (F) t1:ALMIN -> NORMAL, T:23.6C
2025-11-21 00:43:14 PDU local0.info snmpd[358]: Sending trap after thermometer status changed
2025-11-21 00:43:15 PDU daemon.info [INPUTS]: (F) din1:ALMAJ -> NORMAL,T:23.2C,H:37%RH
2025-11-21 00:43:21 PDU daemon.info [BACKEND][343]: (F) User "admin" successfully authenticated
from 192.168.0.1:60923
2025-11-21 00:43:40 PDU daemon.info [INPUTS]: (F) din2:ALMAJ -> NORMAL,T:23.5C,H:34%RH
2025-11-21 00:43:40 PDU daemon.info GUARD: Received new alarm from inputs service = NORMAL
2025-11-21 00:44:27 PDU local1.info [BACKEND]: (F) ACS: Door "Door 1" was added by PDU user
"admin"
2025-11-21 00:44:27 PDU local1.info [BACKEND]: (F) ACS: Door "Door 1" was added by PDU user
"admin"
2025-11-21 00:44:27 PDU local1.info [ACS]: Applying ACS settings
2025-11-21 00:44:27 PDU local1.info [ACS]: (F) ACS: Door "Door 1" is closed
2025-11-21 00:44:38 PDU local1.info [BACKEND]: (F) ACS: Config of zone "zone" was changed by
PDU user "admin"
2025-11-21 00:44:38 PDU local1.info [ACS]: Applying ACS settings
2025-11-21 00:44:40 PDU daemon.info [EXTDEV]: (F) lock: Card 494292636(Mifare Classic),Ivan
successfully added
```

2025-11-21 00:44:40 PDU local1.info [ACS]: (F) ACS: Card 494292636(Mifare Classic),Ivan was added to device "lock"

2025-11-21 00:44:40 PDU daemon.info [EXTDEV]: (F) lock: Card 3925920955(Mifare Classic),Dmitriy successfully added

2025-11-21 00:44:40 PDU local1.info [ACS]: (F) ACS: Card 3925920955(Mifare Classic),Dmitriy was added to device "lock"

2025-11-21 00:45:04 PDU daemon.info [EXTDEV]: (F) lock: unlock by card Dmitriy#3925920955(Mifare Classic)

2025-11-21 00:45:04 PDU daemon.info [EXTDEV]: (F) lock: Unblocked

2025-11-21 00:45:04 PDU local1.info [ACS]: (F) ACS: User "Dmitriy" unlocked the door "Door 1" by card #3925920955

2025-11-21 00:45:04 PDU local0.info snmpd[358]: Sending trap after rem lock status changed

2025-11-21 00:45:13 PDU daemon.info [EXTDEV]: (F) lock: Blocked

2025-11-21 00:45:14 PDU local0.info snmpd[358]: Sending trap after rem lock status changed

2025-11-21 00:45:48 PDU daemon.info [EXTDEV]: (F) lock:NORMAL -> ALMIN, Connection Lost

2025-11-21 00:45:49 PDU local1.info [ACS]: (F) ACS: Opening sensor of door "Door 1" is broken

2025-11-21 00:45:49 PDU local0.info snmpd[358]: Sending trap after rem lock status changed

2025-11-21 00:45:57 PDU daemon.info [EXTDEV]: (F) lock:ALMIN -> NORMAL, Blocked

2025-11-21 00:45:58 PDU local1.info [ACS]: (F) ACS: Door "Door 1" is closed

2025-11-21 00:45:58 PDU local0.info snmpd[358]: Sending trap after rem lock status changed

## 2.6.2. Получить полный журнал событий

**Описание:** получает журнал событий, вместе с архивированными журналами (см 2.6.1)

При наличии объемных архивированных журналов, выполнение запроса может занимать большое количество времени. При выполнении происходят операции записи во FLASH память, поэтому не рекомендуется часто вызывать данный запрос. Для частого получения последних записей в журнале событий рекомендуется использовать запрос 2.6.1, либо протокол syslog (см. Руководство по эксплуатации).

**Endpoint:** /getlog.fcgi

**Тип:** GET

**Формат ответа в случае успешного выполнения:** аналогично 2.6.1

## 2.7. Запросы для мониторинга состояния СКУД

### Статусы датчиков дверей:

- "closed" - Дверь закрыта
- "opened" - Дверь открыта
- "absent" - Датчик двери не назначен
- "broken" - Датчик двери неисправен

### Статусы замков:

- "locked" - Замок заблокирован
- "unlocked" - Замок разблокирован
- "absent" - Устройство замка не назначено
- "broken" - Связь с замком потеряна, либо замок неисправен

### 2.7.1. Мониторинг одной двери СКУД

**Описание:** возвращает состояние двери СКУД по имени или номеру в настройках

**Endpoint:** /acs\_door\_mon.fcgi

**Тип:** GET

**Query параметры:**

- num – номер двери (начиная с 1)
- name – имя двери

Примечание: передается либо параметр num, либо параметр name

**Формат ответа в случае успешного выполнения:**

```
{
  "name": "Main Entrance",
  "sensor_state": "closed",
  "lock_state": "locked",
  "status": 0,
  "broken": false,
  "lastSwipedCard": {
    "id": 123456789,
    "cardType": "mifare_classic",
    "success": true,
    "timestamp": 1715769600
  },
  "lastUnlockTimestamp": 1715769600,
  "dataTimestamp": 1715769605
}
```

**Описание параметров в ответе на запрос:**

- name – Имя двери
- sensor\_state – Состояние датчика двери
- lock\_state - Состояние замка
- status – Статус двери
- broken – Флаг зафиксированного взлома
- lastSwipedCard – Информация о последней приложенной карте
- id – ID последней приложенной карты
- cardType – Тип последней приложенной карты (mifare\_classic, mifare\_plus, em\_marine, hid\_prox)
- success – Успешный (true) проход по последней приложенной карте или нет (false)
- timestamp – Время последнего прикладывания карты в Unix-эпохах
- lastUnlockTimestamp – Время последней разблокировки двери в Unix-эпохах
- dataTimestamp – Время последнего обновления данных о двери в Unix-эпохах\

**Значения поля sensor\_state (Состояния датчика двери):**

- "closed" – Дверь закрыта
- "opened" – Дверь открыта
- "absent" – Датчик двери не назначен
- "broken" – Датчик двери неисправен (**не путать с полем broken у объекта двери**)

**Значения поля lock\_state (Состояние замка):**

- "locked" – Замок заблокирован
- "unlocked" – Замок разблокирован
- "absent" – Устройство замка не назначено
- "broken" – Связь с устройством замка потеряна, либо замок неисправен

**Значения поля status (Статус двери):**

- 0 – NORMAL. Дверь в норме
- 1 – ALMIN. Неисправность датчика двери либо устройства замка
- 2 – ALMAJ. Зафиксирован взлом двери

**Взлом двери** фиксируется, когда происходит открытие двери при заблокированном замке. При обнаружении взлома, поле **broken** принимает значение true и остается таким до квитирования аварии (нажатие кнопки "Квитировать" в WEB интерфейсе), либо отправки GET запроса resetalarm.fcgi, описанного в 2.10.1, либо до разблокировки этой двери картой/паролем.

**Примеры запроса:**

```
curl -X GET -H "Authorization: Bearer jwt" "http://192.168.0.254/acs_door_mon.fcgi?name=Main%20Entrance"
```

```
curl -X GET -H "Authorization: Bearer jwt" "http://192.168.0.254/acs_door_mon.fcgi?num=1"
```

## 2.7.2. Мониторинг всех дверей СКУД

**Описание:** возвращает состояние всех дверей в СКАУД

**Endpoint:** /acs\_door\_mon.fcgi

**Тип:** GET

**Query параметры:** отсутствуют

**Код ответа:** аналогично 2.7.1

**Формат ответа в случае успешного выполнения:**

```
{
  "doors": [
    {
      "name": "Main Entrance",
      "sensor_state": "closed",
      "lock_state": "locked",
      "status": 0,
      "broken": false,
      "lastSwipedCard": {
        "id": 123456789,
        "cardType": "mifare_classic",
        "success": true,
        "timestamp": 1715769600
      },
      "lastUnlockTimestamp": 1715769600,
      "dataTimestamp": 1715769605
    },
    {
      "name": "Server room",
```

```

    "sensor_state": "opened",
    "lock_state": "unlocked",
    "status": 0,
    "broken": false,
    "lastSwipedCard": {
      "id": 453463475,
      "cardType": "em_marine",
      "success": false,
      "timestamp": 1715769441
    },
    "lastUnlockTimestamp": 171576012,
    "dataTimestamp": 1715769606
  }
]
}

```

**Описание параметров в ответе на запрос: 2.7.1**

**Пример запроса:**

```
curl -X GET -H "Authorization: Bearer jwt" "http://192.168.0.254/acs_door_mon.fcgi"
```

### 2.7.3. Получить журнал событий СКУД

**Описание:** получает журнал событий СКУД. Максимальный объем журнала, который может быть получен через этот запрос – 1 мегабайт. Если объем журнала превышает 1 мегабайт, журнал начинает формироваться заново, старый журнал архивируется. Полный журнал (вместе с архивными журналами) можно получить через запрос, описанный в пункте 0. Описание всех событий, которые заносятся в журнал СКУД, приведено в документе "P

**Endpoint:** /acs\_log.fcgi

**Тип:** GET

**Формат ответа в случае успешного выполнения:**

2026-02-09 13:02:56 PDU local1.info [ACS]: (F) ACS: User "Ivan Michailovich" swiped card #494292636(Mifare Classic), reader "AA39Lock". Access is allowed

2026-02-09 13:02:56 PDU local1.info [ACS]: (F) ACS: User "Ivan Michailovich" unlocked the door "AA39DoorH" by card #494292636(Mifare Classic)

2026-02-09 13:02:56 PDU local1.info [ACS]: (F) ACS: Door "AA39DoorH" is unlocked

2026-02-09 13:03:03 PDU local1.info [ACS]: (F) ACS: Door "AA39DoorH" is opened

2026-02-09 13:03:11 PDU local1.info [ACS]: (F) ACS: Door "AA39DoorH" is closed

2026-02-09 13:03:12 PDU local1.info [ACS]: (F) ACS: Door "AA39DoorH" is locked

2026-02-09 13:03:17 PDU local1.info [ACS]: (F) ACS: User "Ivan Michailovich" swiped card #494292636(Mifare Classic), reader "AB39Lock". Access is allowed

2026-02-09 13:03:17 PDU local1.info [ACS]: (F) ACS: User "Ivan Michailovich" unlocked the door "AB39DoorH" by card #494292636(Mifare Classic)

2026-02-09 13:03:17 PDU local1.info [ACS]: (F) ACS: Door "AB39DoorH" is unlocked

2026-02-09 13:03:26 PDU local1.info [ACS]: (F) ACS: Door "AB39DoorH" is locked

2026-02-09 13:03:29 PDU local1.info [ACS]: (F) ACS: Door "AB39DoorH" is unlocked

2026-02-09 13:03:31 PDU local1.info [ACS]: (F) ACS: Door "AB39DoorH" is locked

2026-02-09 13:03:34 PDU local1.info [ACS]: (F) ACS: Door "AB39DoorH" is opened

2026-02-09 13:03:34 PDU local1.info [ACS]: (F) ACS: Door "AB39DoorH" is broken

2026-02-09 13:03:38 PDU local1.info [ACS]: (F) ACS: Door "AB39DoorH" is closed

2026-02-09 13:03:42 PDU local1.info [ACS]: ACS: Alarm quit request received

2026-02-09 13:03:42 PDU local1.info [ACS]: (F) ACS: Door "AB39DoorH" is not broken

2026-02-09 13:03:48 PDU local1.info [ACS]: (F) ACS: User "Ivan Michailovich" entered password, reader "AA39HMI". Access is allowed

2026-02-09 13:03:48 PDU local1.info [ACS]: (F) ACS: User "Ivan Michailovich" unlocked the door group "GroupH21" by password

2026-02-09 13:03:50 PDU local1.info [ACS]: (F) ACS: Door "AA39DoorH" is unlocked

2026-02-09 13:03:50 PDU local1.info [ACS]: (F) ACS: Door "AB39DoorH" is unlocked

2026-02-09 13:03:58 PDU local1.info [ACS]: (F) ACS: Door "AA39DoorH" is locked

2026-02-09 13:03:58 PDU local1.info [ACS]: (F) ACS: Door "AB39DoorH" is locked

2026-02-09 13:04:02 PDU local1.info [ACS]: (F) ACS: Unknown card #3150708959(Mifare Classic) was swiped, reader "AA39Lock"

2026-02-09 13:04:16 PDU local1.info [ACS]: (F) ACS: Unknown password was entered, reader "AA39HMI"

2026-02-09 13:04:54 PDU local1.info [BACKEND]: (F) ACS: User "Stepan stepanovich" was added by PDU user "admin"

2026-02-09 13:04:54 PDU local1.info [ACS]: Applying ACS settings

2026-02-09 13:05:00 PDU local1.info [ACS]: (F) ACS: User "Stepan stepanovich" swiped card #3150708981(Mifare Classic), reader "AB39Lock". Access is denied

#### 2.7.4. Получить полный журнал событий СКУД

**Описание:** получает журнал событий СКУД, вместе с архивированными журналами (см 2.7.3)

При наличии объемных архивированных журналов, выполнение запроса может занимать большое количество времени. При выполнении происходят операции записи во FLASH память, поэтому не рекомендуется часто вызывать данный запрос. Для частого получения последних записей в журнале событий СКУД, рекомендуется использовать запрос 2.7.3, протокол syslog, либо SNMP trap сервер (см. Руководство по эксплуатации).

**Endpoint:** /acs\_log\_archived.fcgi

**Тип:** GET

**Формат ответа в случае успешного выполнения:** аналогично 2.7.3

## 2.8. Запросы для настройки СКУД

### 2.8.1. Получить список устройств считывателей для СКУД

**Описание:** получает список устройств, которые могут быть выбраны как считыватель у Двери или Группы дверей в СКУД

**Endpoint:** /acs\_reader\_devices.fcgi

**Тип:** GET

**Формат ответа в случае успешного выполнения:**

```
{
  "reader_devices": [
    {"name": "REM-lock-1", "id": 1},
    {"name": "HMI-display", "id": 2}
  ]
}
```

### 2.8.2. Получить список устройств замков для СКУД

**Описание:** получает список устройств, которые могут быть выбраны как замок у Двери в СКУД

**Endpoint:** /acs\_lock\_devices.fcgi

**Тип:** GET

**Формат ответа в случае успешного выполнения:**

```
{
  "lock_devices": [
    {"name": "REM-lock-1", "id": 0},
    {"name": "REM-lock-2", "id": 3},
    {"name": "Wiegand-lock", "id": 105}
  ]
}
```

### 2.8.3. Получить список устройств датчиков открытия для СКУД

**Описание:** получает список устройств, которые могут быть выбраны как датчик открытия у Двери в СКУД

**Endpoint:** /acs\_doorsens\_devices.fcgi

**Тип:** GET

**Формат ответа в случае успешного выполнения:**

```
{
  "doorsens_devices": [
    {"name": "REM-lock-1", "id": 0},
    {"name": "REM-lock-2", "id": 3},
    {"name": "DIN 1", "id": 106},
    {"name": "DIN 2", "id": 107},
  ]
}
```

### 2.8.4. Получить список устройств кнопок выхода для СКУД

**Описание:** получает список устройств, которые могут быть прикреплены кнопка выхода у Двери в СКУД

**Endpoint:** /acs\_button\_devices.fcgi

**Тип:** GET

**Формат ответа в случае успешного выполнения:**

```
{
  "button_devices": [
    {"name": "DIN 1", "id": 106},
    {"name": "DIN 2", "id": 107}
  ]
}
```

### 2.8.5. Получить список устройств для двухфакторной аутентификации

**Описание:** получает список устройств, которые могут быть выбраны как устройство двухфакторной аутентификации

**Endpoint:** /devices\_for\_2FA.fcgi

**Тип:** GET

**Формат ответа в случае успешного выполнения:**

```
{
  "devices_2FA": ["Blocker", "Ain3", "Some name"]
}
```

### 2.8.6. Получить список дверей в СКУД

**Описание:** получает список имен дверей в СКУД

**Endpoint:** /acs\_doors.fcgi

**Тип:** GET

**Код ответа:**

- 200 OK — запрос выполнен успешно.

**Формат ответа в случае успешного выполнения:**

```
{
  "doors": ["Door 1", "Door 2", "Door 3", "Door 4"]
}
```

### 2.8.7. Добавить новую дверь в СКУД

**Описание:** добавляет новую дверь в СКУД

**Endpoint:** /acs\_door\_add.fcgi

**Тип:** POST

**Тело запроса:** Тело запроса должно содержать JSON в формате:

```
{
  "name": "Door 1",
  "reader_ids": [0, 1],
  "lock_id": 0,
  "door_sensor": 0,
  "device2FA": "Blocker",
  "has_button": true,
  "button_id": 102
}
```

**Описание параметров в теле запроса:**

- name – имя добавляемой двери. Имя двери должно быть уникально.
- reader\_ids – ID устройств считывателей, закрепляемых за дверью
- lock\_id – ID устройства замка
- door\_sensor – ID устройства датчика открытия двери
- device2FA – Имя устройства для двухфакторной аутентификации. Если двухфакторная аутентификация не используется, значение должно быть пустым: "device2FA": ""

- has\_button – флаг наличия кнопки выхода.
- button\_id – ID устройства кнопки выхода. Опционален. Если has\_button == false, параметр button\_id не обязателен и игнорируется при запросе

### 2.8.8. Удалить дверь из СКУД

**Описание:** удаляет дверь из СКУД

**Endpoint:** /acs\_door\_delete.fcgi

**Тип:** POST

**Query параметры:**

- num – номер двери в списке дверей в СКУД. Начинается с 1.
- name – имя двери

Примечание: в запрос передается либо параметр num, либо параметр name

### 2.8.9. Получить настройки двери в СКУД

**Описание:** получает настройки двери в СКУД

**Endpoint:** /acs\_door\_config.fcgi

**Тип:** GET

**Query параметры:**

- num – номер двери в списке дверей в СКУД. Начинается с 1.
- name – имя двери

Примечание: в запрос передается либо параметр num, либо параметр name

**Формат ответа в случае успешного выполнения:**

```
{
  "name": "Door 1",
  "reader_ids": [0, 1],
  "lock_id": 0,
  "door_sensor": 0,
  "device2FA": "Blocker",
  "has_button": true,
  "button_id": 102
}
```

**Описание параметров в ответе на запрос:**

- name – имя двери
- reader\_ids – ID устройств считывателей, закрепленных за дверью
- lock\_id – ID устройства замка
- door\_sensor – ID устройства датчика открытия двери
- device2FA – Имя устройства для двухфакторной аутентификации. Если двухфакторная аутентификация не используется, значение будет пустым: "device2FA": ""
- has\_button – флаг наличия кнопки выхода.
- button\_id – ID устройства кнопки выхода. Опционален. Если has\_button == false, параметр button\_id будет отсутствовать в ответе

### 2.8.10. Записать настройки двери в СКУД

**Описание:** записывает настройки двери в СКУД

**Endpoint:** /acs\_door\_config.fcgi

**Тип:** POST

**Query параметры:**

- num – номер двери в списке дверей в СКУД. Начинается с 1.

- name – имя двери

Примечание: в запрос передается либо параметр num, либо параметр name

**Тело запроса:** Тело запроса должно содержать JSON в формате:

```
{
  "name": "Door 1",
  "reader_ids": [0, 1],
  "lock_id": 0,
  "door_sensor": 0,
  "device2FA": "Blocker",
  "has_button": true,
  "button_id": 102
}
```

**Описание параметров в теле запроса:**

- name – имя двери. Имя двери должно быть уникально.
- reader\_ids – ID устройств считывателей, закрепляемых за дверью
- lock\_id – ID устройства замка
- door\_sensor – ID устройства датчика открытия двери
- device2FA – Имя устройства для двухфакторной аутентификации. Если двухфакторная аутентификация не используется, значение должно быть пустым: "device2FA": ""
- has\_button – флаг наличия кнопки выхода.
- button\_id – ID устройства кнопки выхода. Опционален. Если has\_button == false, параметр button\_id не обязателен и игнорируется при запросе

### 2.8.11. Получить список групп дверей в СКУД

**Описание:** получает список имен групп дверей в СКУД

**Endpoint:** /acs\_doorgroups.fcgi

**Тип:** GET

**Формат ответа в случае успешного выполнения:**

```
{
  "door_groups": ["Group 1", "Group 2", "Group 3"]
}
```

### 2.8.12. Записать настройки двери в СКУД

**Описание:** добавляет новую группу дверей в СКУД

**Endpoint:** /acs\_doorgroup\_add.fcgi

**Тип:** POST

**Тело запроса:** Тело запроса должно содержать JSON в формате:

```
{
  "name": "Group 1",
  "doors": ["Front door", "Back door", "Door 3"],
  "reader_ids": [ 1, 3 ]
}
```

**Описание параметров в теле запроса:**

- name – имя добавляемой группы дверей. Имя группы дверей должно быть уникально
- doors – список имен дверей в группе
- reader\_ids – ID устройств считывателей, закрепляемых за группой дверей

### 2.8.13. Удалить группу дверей из СКУД

**Описание:** удаляет группу дверей из СКУД

**Endpoint:** /acs\_doorgroup\_delete.fcgi

**Тип:** POST

**Query параметры:**

- num – номер группы дверей в списке групп дверей. Начинается с 1.
- name – имя группы дверей

Примечание: в запрос передается либо параметр num, либо параметр name

#### 2.8.14. Получить настройки группы дверей в СКУД

**Описание:** получает настройки группы дверей в СКУД

**Endpoint:** /acs\_doorgroup\_config.fcgi

**Тип:** GET

**Query параметры:**

- num – номер группы дверей в списке групп дверей. Начинается с 1.
- name – имя группы двери

Примечание: в запрос передается либо параметр num, либо параметр name

**Формат ответа в случае успешного выполнения:**

```
{
  "name": "Group 1",
  "doors":["Front door", "Back door", "Door 3"],
  "reader_ids":[ 1, 3 ]
}
```

**Описание параметров в ответе на запрос:**

- name – имя группы дверей
- doors – список имен дверей в группе
- reader\_ids – ID устройств считывателей, закрепленных за группой дверей

#### 2.8.15. Записать настройки группы дверей в СКУД

**Описание:** записывает настройки группы дверей в СКУД

**Endpoint:** /acs\_doorgroup\_config.fcgi

**Тип:** POST

**Query параметры:**

- num – номер группы дверей в списке группы дверей. Начинается с 1.
- name – имя группы дверей

Примечание: в запрос передается либо параметр num, либо параметр name

**Тело запроса:** Тело запроса должно содержать JSON в формате:

```
{
  "name": "Group 1",
  "doors":["Front door", "Back door", "Door 3"],
  "reader_ids":[ 1, 3 ]
}
```

**Описание параметров в теле запроса:**

- name – имя группы дверей
- doors – список имен дверей в группе
- reader\_ids – ID устройств считывателей, закрепленных за группой дверей

### 2.8.16. Получить список зон доступа в СКУД

**Описание:** получает список имен зон доступа в СКУД

**Endpoint:** /acs\_zones.fcgi

**Тип:** GET

**Формат ответа в случае успешного выполнения:**

```
{
  "zones ": ["Zone 1", "Zone 2", "Zone 3"]
}
```

### 2.8.17. Добавить новую зону доступа в СКУД

**Описание:** добавляет новую зону доступа в СКУД

**Endpoint:** /acs\_zone\_add.fcgi

**Тип:** POST

**Тело запроса:** Тело запроса должно содержать JSON в формате:

```
{
  "name": "Zone 2",
  "doors":["Front door", "Back door"],
  "door_groups":["Common doors", "Private doors"]
}
```

**Описание параметров в теле запроса:**

- name – имя добавляемой зоны доступа. Имя зоны доступа должно быть уникально
- doors – список имен дверей в зоне доступа
- door\_groups – список имен групп дверей в зоне доступа

### 2.8.18. Удалить зону доступа из СКУД

**Описание:** удаляет зону доступа из СКУД

**Endpoint:** /acs\_zone\_delete.fcgi

**Тип:** POST

**Query параметры:**

- num – номер зоны доступа в списке зон доступа. Начинается с 1.
- name – имя зоны доступа

Примечание: в запрос передается либо параметр num, либо параметр name

### 2.8.19. Получить настройки зоны доступа в СКУД

**Описание:** получает настройки зоны доступа в СКУД

**Endpoint:** /acs\_zone\_config.fcgi

**Тип:** GET

**Query параметры:**

- num – номер зоны доступа в списке зон доступа. Начинается с 1.
- name – имя зоны доступа

Примечание: в запрос передается либо параметр num, либо параметр name

**Формат ответа в случае успешного выполнения:**

```
{
  "name": "Zone 2",
  "doors":["Front door", "Back door"],
  "door_groups":["Common doors", "Private doors"]
}
```

**Описание параметров в ответе на запрос:**

- name – имя зоны доступа
- doors – список имен дверей в зоне доступа
- door\_groups – список имен групп дверей в зоне доступа

**2.8.20. Записать настройки зоны доступа в СКУД**

**Описание:** записывает настройки зоны доступа в СКУД

**Endpoint:** /acs\_zone\_config.fcgi

**Тип:** POST

**Query параметры:**

- num – номер зоны доступа в списке зон доступа. Начинается с 1.
- name – имя зоны доступа

Примечание: в запрос передается либо параметр num, либо параметр name

**Тело запроса:** Тело запроса должно содержать JSON в формате:

```
{
  "name": "Zone 2",
  "doors":["Front door", "Back door"],
  "door_groups":["Common doors", "Private doors"]
}
```

**Описание параметров в теле запроса:**

- name – имя зоны доступа
- doors – список имен дверей в зоне доступа
- door\_groups – список имен групп дверей в зоне доступа

**2.8.21. Получить доступные двери для зон доступа в СКУД**

**Описание:** получает для всех зон доступа доступные двери. В настройках зоны доступа указываются включенные в нее двери и группы дверей (в которые включены другие двери). Чтобы легче было получить полный список всех дверей, которые включены в зону доступа, можно воспользоваться этим методом.

**Endpoint:** /acs\_doors\_in\_zones.fcgi

**Тип:** GET

**Формат ответа в случае успешного выполнения:**

```
{
  "zones": [
    {"name":"Private zone", "doors":["Door 1", "Door 2", "Door 3", "Door 4"]},
    {"name":"Common zone", "doors":["Door 1", "Door 2"]}
  ]
}
```

**2.8.22. Получить список пользователей в СКУД**

**Описание:** получает список имен пользователей в СКУД

**Endpoint:** /acs\_users.fcgi

**Тип:** GET

**Формат ответа в случае успешного выполнения:**

```
{
  "users": ["Ivan", "Dmitriy", "Denis"]
}
```

**2.8.23. Добавить нового пользователя в СКУД**

**Описание:** добавляет нового пользователя в СКУД

**Endpoint:** /acs\_user\_add.fcgi

**Тип:** POST

**Тело запроса:** Тело запроса должно содержать JSON в формате:

```
{
  "name": "Ivan",
  "password": "1234",
  "card_id": 8837372,
  "card_type": "mifare_classic",
  "zones": ["Zone 1", "Zone 2"]
}
```

**Описание параметров в теле запроса:**

- name – имя добавляемого пользователя. Должно быть уникально.
- password – пароль для ввода на дисплеях и клавиатурах. Должен быть уникальным. Если пользователь не должен открывать двери с помощью ввода пароля, значение можно оставить пустым, "password": ""
- card\_id – номер карточки
- card\_type – тип карты (mifare\_classic, mifare\_plus, em\_marine)
- zones – список имен зон доступа, в которые добавлен пользователь

#### 2.8.24. Удалить пользователя из СКУД

**Описание:** удаляет зону доступа из СКУД

**Endpoint:** /acs\_user\_delete.fcgi

**Тип:** POST

**Query параметры:**

- num – номер пользователя в списке пользователей СКУД. Начинается с 1.
- name – имя пользователя

Примечание: в запрос передается либо параметр num, либо параметр name

#### 2.8.25. Получить настройки пользователя в СКУД

**Описание:** получает настройки пользователя в СКУД

**Endpoint:** /acs\_user\_config.fcgi

**Тип:** GET

**Query параметры:**

- num – номер пользователя в списке пользователей СКУД. Начинается с 1.
- name – имя пользователя

Примечание: в запрос передается либо параметр num, либо параметр name

**Формат ответа в случае успешного выполнения:**

```
{
  "name": "Ivan",
  "password": "1234",
  "card_id": 8837372,
  "card_type": "mifare_plus",
  "zones": ["Zone 1", "Zone 2"]
}
```

**Описание параметров в ответе на запрос:**

- name – имя пользователя.
- password – пароль для ввода на дисплеях и клавиатурах. Если пользователь не может открывать двери с помощью ввода пароля, имеет пустое значение "password": ""

- card\_id – номер карточки
- card\_type – тип карты (mifare\_classic, mifare\_plus, em\_marine)
- zones – список имен зон доступа, в которые добавлен пользователь

### 2.8.26. Записать настройки пользователя в СКУД

**Описание:** записывает настройки пользователя в СКУД

**Endpoint:** /acs\_user\_config.fcgi

**Тип:** POST

**Query параметры:**

- num – номер пользователя в списке пользователей СКУД. Начинается с 1.
- name – имя пользователя

Примечание: в запрос передается либо параметр num, либо параметр name

**Тело запроса:** Тело запроса должно содержать JSON в формате:

```
{
  "name": "Ivan",
  "password": "1234",
  "card_id": 8837372,
  "card_type": "em_marine",
  "zones": ["Zone 1", "Zone 2"]
}
```

**Описание параметров в теле запроса:**

- name – имя пользователя. Должно быть уникальным
- password – пароль для ввода на дисплеях и клавиатурах. Должен быть уникальным. Если пользователь не должен открывать двери с помощью ввода пароля, значение можно оставить пустым, "password": ""
- card\_id – номер карточки
- card\_type – тип карты (mifare\_classic, mifare\_plus, em\_marine)
- zones – список имен зон доступа, в которые добавлен пользователь

### 2.8.27. Экспорт настроек СКУД

**Описание:** Возвращает файл с настройками СКУД, а также настройками внутренних и внешних устройств.

**Endpoint:** /acs\_export.fcgi

**Тип:** GET

### 2.8.28. Импорт настроек СКУД

**Описание:** Загружает и применяет настройки СКУД, внутренних и внешних устройств из файла, полученного при экспорте настроек.

**Endpoint:** /acs\_import.fcgi

**Тип:** POST

**Query параметры:**

- withusers (опциональный) - включить/исключить пользователей (0/1, по умолчанию 0)

**Form Data:**

- acs.sett (обязательный) - файл с настройками СКУД в формате multipart/form-data

### 2.8.29. Очистить журнал событий СКУД

**Описание:** очищает журнал событий СКУД

**Endpoint:** /acs\_log\_clear.fcgi

**Тип:** POST

## 2.9. Запросы для экспорта и импорта настроек

### 2.9.1. Экспорт настроек

**Описание:** Возвращает файл с выбранными категориями настроек

**Endpoint:** /config\_export.fcgi

**Тип:** GET

**Query параметры (выбор категорий настроек для экспорта):**

- devicesInputsAndACS – настройки внутренних, внешних устройств и СКУД
- power – настройки розеток, встроенного ABP, порогов индикации контуров
- protocols – настройки протоколов (RADIUS, SNMP, NTP, и т.д)
- scripts – скрипты гибкой логики
- serial – настройки интерфейсов
- network – настройки сети

Примечание: для включения определенной категории настроек в выходной файл, нужно добавить соответствующий этой категории query параметр со значением 1 (например, power=1).

**Пример запроса:**

```
curl -X GET -H "Authorization: Bearer jwt" http://192.168.0.254/config_export.fcgi?devicesInputsAndACS=1&power=1&protocols=1 -o config.set
```

### 2.9.2. Экспорт настроек на TFTP

**Описание:** Загружает файл с выбранными категориями настроек на TFTP сервер, указанный в настройках протоколов

**Endpoint:** /config\_export\_tftp.fcgi

**Тип:** GET

**Query параметры (выбор категорий настроек для экспорта):**

- devicesInputsAndACS – настройки внутренних, внешних устройств и СКУД
- power – настройки розеток, встроенного ABP, порогов индикации контуров
- protocols – настройки протоколов (RADIUS, SNMP, NTP, и т.д)
- scripts – скрипты гибкой логики
- serial – настройки интерфейсов
- network – настройки сети
- filename – имя файла для экспорта на TFTP сервер

Примечание: для включения определенной категории настроек в выходной файл, нужно добавить соответствующий этой категории query параметр со значением 1 (например, power=1)

**Пример запроса:**

```
curl -X GET -H "Authorization: Bearer jwt" http://192.168.0.254/config_export_tftp.fcgi?devicesInputsAndACS=1&power=1&filename=config.set
```

### 2.9.3. Импорт настроек

**Описание:** импортирует выбранные категории настроек из файла

**Endpoint:** /config\_import.fcgi

**Тип:** GET

**Query параметры (выбор категорий настроек для импорта):**

- devicesInputsAndACS – настройки внутренних, внешних устройств и СКУД
- power – настройки розеток, встроенного АБП, порогов индикации контуров
- protocols – настройки протоколов (RADIUS, SNMP, NTP, и т.д)
- scripts – скрипты гибкой логики
- serial – настройки интерфейсов
- network – настройки сети. **При импорте настроек сети потребуется перезагрузка контроллера. Контроллер будет доступен по новому адресу после перезагрузки.**

Примечание: для импорта определенной категории настроек, нужно добавить соответствующий этой категории query параметр со значением 1 (например, power=1).

**Form Data:**

- sett (обязательный) - файл с настройками (полученный при экспорте).

**Пример запроса:**

```
curl -X GET -H "Authorization: Bearer jwt"  
http://192.168.0.254/config_import.fcgi?devicesInputsAndACS=1&power=1 -F "sett=@./config.set"
```

### 2.9.4. Импорт настроек с TFTP сервера

**Описание:** импортирует выбранные категории настроек из файла, лежащим на TFTP сервере

**Endpoint:** /config\_import\_tftp.fcgi

**Тип:** GET

**Query параметры (выбор категорий настроек для импорта):**

- devicesInputsAndACS – настройки внутренних, внешних устройств и СКУД
- power – настройки розеток, встроенного АБП, порогов индикации контуров
- protocols – настройки протоколов (RADIUS, SNMP, NTP, и т.д)
- scripts – скрипты гибкой логики
- serial – настройки интерфейсов
- network – настройки сети. **При импорте настроек сети потребуется перезагрузка контроллера. Контроллер будет доступен по новому адресу после перезагрузки.**
- filename – имя файла на TFTP сервере

Примечание: для импорта определенной категории настроек, нужно добавить соответствующий этой категории query параметр со значением 1 (например, power=1).

**Пример запроса:**

```
curl -X GET -H "Authorization: Bearer jwt"  
http://192.168.0.254/config_import_tftp.fcgi?devicesInputsAndACS=1&power=1&filename=config.set
```

## 2.10. Другие запросы

### 2.10.1. Квитирование аварии

**Описание:** Сбрасывает состояние аварии на контроллере

**Endpoint:** /resetalarm.fcgi

**Тип:** GET

### 2.10.2. Перезагрузка контроллера

**Описание:** Перезагружает контроллер

**Endpoint:** /reboot.fcgi

**Тип:** GET