

Содержание

1. Общее положение	3
2. Информация об устройстве и настройка времени	5
2.1 Получение информации о контроллере	5
2.2 Получение времени контроллера	6
2.3 Установка времени контроллера	7
3 Журнал событий	7
3.1 Получение количества событий контроллера	7
3.2 Выгрузка событий контроллера	8
4 Настройка расписаний доступов	10
4.1 Получение расписаний доступов	10
4.2 Добавление и изменение расписаний доступов	12
4.3 Удаление расписаний доступов	14
5 Управление пользователями	14
5.1 Получение пользователей	14
5.2 Добавление пользователей	16
5.3 Удаление пользователей	17
6 Управление и настройка контроллера	18
6.1 Управление точками прохода	18
6.2 Установка режима прохода	19
6.3 Очистка контроллера	19
6.3.1 Очистка пользователей	20
6.3.2 Очистка расписаний	20
6.3.3 Очистка событий	21
6.3.4 Очистка логов	21
6.3.5 Полная очистка	22
6.4 Формат идентификатора карты	22
6.4.1 Получение формата идентификатора карты	22
6.4.2 Установка формата идентификатора карты	23

1. Общее положение

Открытое API контроллера доступно начиная с версии прошивки 1.0.130.

Взаимодействие с API осуществляется исключительно по защищённому протоколу HTTPS на порту 8443. По умолчанию используется самоподписанный TLS сертификат. HTTP запросы игнорируются. Контроллер позволяет через web-интерфейс установить пользовательский сертификат. Тело запроса должно быть оформлено в формате JSON. При успешной обработке запроса тело ответа также содержит данные в формате JSON.

Авторизация:

Для доступа к API используется дайджест-аутентификация с алгоритмом SHA-256. Учётные данные (логин и пароль) совпадают с авторизационными данными веб-интерфейса.

***Важно:** после обновления с прошивки версии ниже 1.0.130 необходимо повторно задать имя пользователя и пароль в настройках веб-интерфейса (даже если они не изменились).*

Методы запросов:

Для взаимодействия с API используются стандартные HTTP-методы. Каждый метод определяет тип операции и задаёт ожидаемое поведение контроллера:

Метод	Описание
GET	Предназначен для получения информации.
PUT	Используется для полной замены состояния.
DELETE	Используется для удаления.
POST	Вызывает выполнение действия.

Состояния ответов:

Контроллер сообщает о результате обработки запроса с помощью стандартных HTTP-статусов. Клиент должен учитывать эти коды при интерпретации ответа:

Код ответа	Описание
200	Запрос успешно обработан, результат возвращён в теле ответа в формате JSON.
400	Тело запроса не соответствует формату JSON.
401	Требуется аутентификация.
404	Метод отсутствует.
405	Использован HTTP-метод, который не поддерживается в данном контексте.
422	Синтаксис JSON корректен, но содержимое не прошло валидацию (например, отсутствует обязательное поле или указано недопустимое значение).
500	Произошла непредвиденная ошибка на стороне контроллера, не связанная с запросом клиента.
501	Неподдерживаемая операция.
503	Использование API запрещено (необходимо отвязать контроллер от сервера).

Формат JSON-запроса:

Все запросы должны быть оформлены в формате JSON и содержать как минимум одно обязательное поле. Дополнительные поля зависят от типа выполняемой команды.

```
{
    "target": "имя команды",
    "extra": любой тип данных* ,
    // ... параметры команды (определяются по значению поля "target")
}
```

1. `target` — определяет, какая команда выполняется или какой источник данных запрашивается.
2. `extra` (необязательное поле) — возвращается в ответе без изменений; используется для сопоставления асинхронных запросов и ответов.

Ключи команды:

Помимо базовых полей, в теле запроса могут присутствовать дополнительные параметры, специфичные для выбранной команды (указанной в `target`). Набор и формат этих параметров зависят от конкретной команды и описаны в соответствующих разделах документации.

2. Информация об устройстве и настройка времени

Данный раздел описывает системные команды для взаимодействия с контроллером, включая получение и установку его внутреннего времени, и другие служебные операции. Для каждого метода API приведены HTTP-метод, путь и структура тела запроса в формате JSON, а также описание обязательных полей.

2.1 Получение информации о контроллере

Метод возвращает общую информацию о контроллере, включая идентификационные данные (производитель, модель, серийный номер, версия прошивки), лицензионные параметры, текущее состояние ресурсов (количество расписаний, событий, идентификаторов и др.), а также максимальные ёмкости по каждому типу данных.

Запрос:

GET <a href="https://<ip_адрес>:8443/api/capability">https://<ip_адрес>:8443/api/capability			
Поле	Тип	Значение	Описание
target	VARCHAR	capability	Запросить общую информацию о контроллере.
<pre>{ "target": "capability" }</pre>			

Ответ:

Поле	Тип	Значение	Описание
vendor	VARCHAR	ControlGate	Производитель устройства.
model	VARCHAR	CG-ACS-R2	Модель устройства.
serial	VARCHAR	16554153cd450b4a	Серийный номер.
version	VARCHAR	1.0.128	Версия прошивки.
license	INTEGER	4	Количество лицензий.
schedules	INTEGER	0	Количество расписаний.
schedulesCapacity	INTEGER	1000	Максимальное количество расписаний.
events	INTEGER	0	Количество событий.
eventsCapacity	INTEGER	200000	Максимальное количество событий.
identify	INTEGER	0	Количество всех идентификаторов.
identifyCapacity	INTEGER	50000	Максимальное количество всех идентификаторов.
cards	INTEGER	0	Количество карт.
persons	INTEGER	0	Количество пользователей.
fingerprints	INTEGER	0	Количество отпечатков.
faces	INTEGER	0	Количество лиц.

gpio	INTEGER	2	Количество встроенных выходов.
osdp	INTEGER	0	Количество OSDP устройств.
<pre>{ "vendor": "ControlGate", "model": "CG-ACS-R2", "serial": "16554153cd450b4a", "version": "1.0.128", "license": 4, "schedules": 0, "schedulesCapacity": 1000, "events": 0, "eventsCapacity": 200000, "identify": 0, "identifyCapacity": 50000, "cards": 0, "persons": 0, "fingerprints": 0, "faces": 0, "gpio": 2, "osdp": 0 }</pre>			

2.2 Получение времени контроллера

Метод возвращает текущее время контроллера в виде Unix-времени — целого числа секунд с начала эпохи Unix (00:00:00 UTC, 1 января 1970 года, GMT±0).

Запрос:

GET <i>https://<ip_адрес>:8443/api/settings</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	time	Запросить время контроллера в UTC GMT±0.
<pre>{ "target": "time" }</pre>			

Ответ:

Поле	Тип	Значение	Описание
time	UINT64	1712345678	Время Unix timestamp в секундах UTC GMT±0.
<pre>{ "time": 1712345678 }</pre>			

2.3 Установка времени контроллера

Метод устанавливает время контроллера в виде Unix-времени — целого числа секунд с начала эпохи Unix (00:00:00 UTC, 1 января 1970 года, GMT±0). Ответ возвращает текущее время контроллера после его установки.

Запрос:

PUT <i>https://<ip_адрес>:8443/api/settings</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	time	Запросить время контроллера в UTC GMT±0.
time	UINT64	1712346532	Время Unix timestamp в секундах UTC GMT±0.
<pre>{ "target": "time", "time": 1712346532 }</pre>			

Ответ:

Поле	Тип	Значение	Описание
time	UINT64	1712346532	Время Unix timestamp в секундах UTC GMT±0.
<pre>{ "time": 1712346532 }</pre>			

3 Журнал событий

Раздел содержит методы для получения событий из журнала контроллера, а также описание типов событий, их кодов и значений.

3.1 Получение количества событий контроллера

Метод возвращает количество событий, зафиксированных контроллером в указанном временном интервале. Ответ возвращает найденное количество событий в памяти устройства.

Запрос:

GET <i>https://<ip_адрес>:8443/api/access/events</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	count	Получить количество событий.
start	UINT64	1712340000	Время начала Unix timestamp в секундах UTC GMT±0. Необязательное поле, по умолчанию 00:00:00 UTC, 1 января 1970 года, GMT±0.
end	UINT64	1712346532	Время конца Unix timestamp в секундах UTC GMT±0. Необязательное поле, по умолчанию время в момент выполнения запроса.

```
{
  "target":"count",
  "start":1712340000,
  "end":1712346532
}
```

Ответ:

Поле	Тип	Значение	Описание
count	INTEGER	10	Количество найденных событий.
<pre>{ "count":10 }</pre>			

3.2 Выгрузка событий контроллера

Метод возвращает список событий, зафиксированных контроллером в указанном временном интервале. Ответ возвращает найденные события в памяти устройства.

Запрос:

GET <i>https://<ip_адрес>:8443/api/access/events</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	events	Получить события.
start	UINT64	1712340000	Время начала Unix timestamp в секундах UTC GMT±0. Необязательное поле, по умолчанию 00:00:00 UTC, 1 января 1970 года, GMT±0.
end	UINT64	1712346532	Время конца Unix timestamp в секундах UTC GMT±0. Необязательное поле, по умолчанию время в момент выполнения запроса.
offset	INTEGER	0	Игнорировать указанное количество событий с начала выборки. Необязательное поле, по умолчанию 0.
limit	INTEGER	10000	Количество событий, возвращаемых в ответе. Необязательное поле, по умолчанию 10000. Максимальное значение 10000.
<pre>{ "target":"events", "start":1712340000, "end":1720013443, "offset":0, "limit":10000 }</pre>			

Ответ:

Поле	Тип	Значение	Описание
count	INTEGER	2	Количество найденных событий.
events	ARRAY	[{...},{...}]	Массив JSON событий.
type	INTEGER	6	Тип события. Список типов событий приведены в Таблице ниже.
time	UINT64	1720013443	Время события Unix timestamp в секундах UTC GMT±0.
person	INTEGER	0	Уникальный номер пользователя.
direction	INTEGER	0	Направление прохода.
channel	INTEGER	5	Номер прохода. Соответствует настройкам прохода в web-интерфейсе контроллера.
device	INTEGER	2	Вспомогательное поле, уточняющее инициатора в некоторых событиях.
record	INTEGER	0	Номер записи.
<pre>{ "count": 2, "events": [{ "type": 6, "time": 1720013443, "person": 0, "direction": 0, "channel": 5, "record": 0 }, { "type": 253, "time": 1720013443, "device": 2, "record": 1 }] }</pre>			

ТИПЫ СОБЫТИЙ

Номер	Описание
4	Дверь оставлена открытой.
5	Сработала пожарная тревога.
6	Подача сигнала о пожарной тревоге завершена.
7	Успешная авторизация прохода с фактом открытия двери.
8	Дверь взломана.
9	Дверь открыта кнопкой.
10	Нарушение расписания прохода.
11	Нарушение режима «Antipassback» (повторный проход).

12	Нарушение режима «Owner» (отсутствует хозяин помещения).
13	Успешная авторизация прохода.
14	Ошибка авторизации прохода.
15	Проход открыт оператором.
16	Изменен режим прохода.
17	Успешная авторизация прохода в режиме «Проход заблокирован».
18	Зафиксировано вскрытие корпуса контроллера.
19	Корпус контроллера восстановлен (закрыт).
20	Успешная авторизация прохода и подтверждение прохода по лазерам замка.
21	Успешная авторизация прохода в режиме «Свободный проход».
200	Подключен сервер по внутреннему протоколу.
201	Отключен сервер по внутреннему протоколу.
253	Вход перешёл в неактивное состояние.
254	Вход перешёл в активное состояние.

4 Настройка расписаний доступов

Раздел содержит методы для управления расписаниями доступа: создание, редактирование и удаление временных правил, определяющих, когда пользователи могут проходить через точки доступа.

4.1 Получение расписаний доступов

Метод возвращает все расписания, заданные в контроллере.

Расписание состоит из двух частей:

1. Годовое расписание определяет, какое дневное расписание применяется к каждому дню года.
2. Дневное расписание задаёт правила доступа в течение суток (например, разрешённые временные интервалы).

В ответе метода возвращаются как годовые, так и дневные расписания.

В приведённом примере расписание настроено так, что доступ разрешён в любое время, то есть без временных ограничений.

Запрос:

GET <i>https://<ip_адрес>:8443/api/access/schedules</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	schedules	Получить расписания.
<pre>{ "target": "schedules" }</pre>			

4.2 Добавление и изменение расписаний доступов

Метод добавляет или изменяет расписания в контроллере. Запрос может содержать:

1. только годовые расписания;
2. только дневные расписания;
3. оба типа одновременно.

Годовое расписание задаётся по месяцам: для каждого дня месяца указывается идентификатор “id” соответствующего дневного расписания. Дневное расписание представляет собой список временных интервалов, в течение которых разрешён доступ. Каждый интервал задаётся временем начала и окончания.

В примере с “Понедельника” по “Пятницу” используется расписание с “id” 1, в “Субботу” расписание с “id” 2, в “Воскресенье” доступ запрещен.

Расписание с “id” 1 разрешает доступ с 08:00 до 12:00 и с 13:00 до 18:00.

Расписание с “id” 2 разрешает доступ с 09:00 до 12:00.

Запрос:

PUT <a href="https://<ip_адрес>:8443/api/access/schedules">https://<ip_адрес>:8443/api/access/schedules			
Поле	Тип	Значение	Описание
target	VARCHAR	schedules	Установить расписание.
years	ARRAY	[{...},{...}]	Массив JSON годовых расписаний.
id	INTEGER	1	Уникальный номер годового расписания.
january - december	ARRAY	[...]	Массив UINT16 уникальных номеров дневных расписаний на каждый день месяца.
days	ARRAY	[{},{}]	Массив JSON дневных расписаний.
id	INTEGER	1,2	Уникальный номер дневного расписания.
day	ARRAY	[{...},{...}]	Массив JSON временных интервалов доступа.
start	VARCHAR	"07:00"	Начало действия доступа.
end	VARCHAR	"12:00"	Окончание действия доступа.
<pre>{ "target":"schedules", "years": [{ "id": 1, "year": { "january": [1,1,1,2,0,1,1,1,1,1,2,0,1,1,1,1,1,2,0,1,1,1,1,1,2,0,1,1,1,1,1,1], "february": [2,0,1,1,1,1,1,2,0,1,1,1,1,1,1,2,0,1,1,1,1,1,2,0,1,1,1,1,1,0,0,0], "march": [2,0,1,1,1,1,1,2,0,1,1,1,1,1,1,2,0,1,1,1,1,1,2,0,1,1,1,1,1,2,0,1], "april": [1,1,1,1,2,0,1,1,1,1,1,2,0,1,1,1,1,1,2,0,1,1,1,1,1,2,0,1,1,1,0], "may": [1,1,2,0,1,1,1,1,1,2,0,1,1,1,1,1,2,0,1,1,1,1,1,2,0,1,1,1,1,1,2], "june": [0,1,1,1,1,1,2,0,1,1,1,1,1,2,0,1,1,1,1,1,2,0,1,1,1,1,1,2,0,1,0], "july": [1,1,1,1,2,0,1,1,1,1,1,2,0,1,1,1,1,1,1,2,0,1,1,1,1,1,2,0,1,1,1,1], } }] }</pre>			

```

    "august": [1,2,0,1,1,1,1,1,2,0,1,1,1,1,2,0,1,1,1,1,2,0],
    "september": [1,1,1,1,1,2,0,1,1,1,1,2,0,1,1,1,1,2,0,1,1,1,1,2,0,1,1,0],
    "october": [1,1,1,2,0,1,1,1,1,2,0,1,1,1,1,2,0,1,1,1,1,2,0,1,1,1,1],
    "november": [2,0,1,1,1,1,1,2,0,1,1,1,1,2,0,1,1,1,1,2,0,1,1,1,1,2,0,0],
    "december": [1,1,1,1,1,2,0,1,1,1,1,1,2,0,1,1,1,1,2,0,1,1,1,1,2,0,1,1,1]
  }
}
],
"days": [
  {
    "id": 1,
    "day": [
      {
        "start": "07:00",
        "end": "12:00"
      },
      {
        "start": "13:00",
        "end": "18:00"
      }
    ]
  },
  {
    "id": 2,
    "day": [
      {
        "start": "09:00",
        "end": "12:00"
      }
    ]
  }
]
}
]
}
}

```

Ответ:

Поле	Тип	Значение	Описание
error	INTEGER	0	Наличие ошибки при выполнении. 0 - успешно выполнено. 1 - ошибка при выполнении.
<pre>{ "error": 0 }</pre>			

4.3 Удаление расписаний доступов

Метод удаляет указанные годовые и/или дневные расписания из контроллера.

В запросе можно передать:

1. только годовые расписания;
2. только дневные расписания;
3. оба типа одновременно.

В ответе возвращается статус выполнения запроса (успех или ошибка).

Запрос:

DELETE <i>https://<ip_адрес>:8443/api/access/schedules</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	schedules	Удалить расписание.
years	ARRAY	[1]	Массив INTEGER уникальных номеров годовых расписаний.
days	ARRAY	[1,2]	Массив UINT16 уникальных номеров дневных расписаний.
<pre>{ "target": "schedules", "years": [1], "days": [1,2] }</pre>			

Ответ:

Поле	Тип	Значение	Описание
error	INTEGER	0	Наличие ошибки при выполнении. 0 - успешно выполнено. 1 - ошибка при выполнении.
<pre>{ "error": 0 }</pre>			

5 Управление пользователями

Раздел содержит методы для управления пользователями: добавление новых, получение списка существующих и удаление записей из контроллера.

5.1 Получение пользователей

Метод возвращает список всех пользователей, загруженных в контроллер, включая их уникальные идентификаторы и права доступа. Если в запросе указан идентификатор конкретного пользователя, метод возвращает данные только этого пользователя.

Запрос:

GET <i>https://<ip_адрес>:8443/api/access/persons</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	persons	Получить пользователей.
person	INTEGER	1	Идентификатор (необязательное поле).
<pre>{ "target": "persons", "person": 1 }</pre>			

Ответ:

Поле	Тип	Значение	Описание
count	INTEGER	1	Количество найденных пользователей.
persons	ARRAY	[{}]	Массив JSON пользователей.
id	INTEGER	1	Уникальный номер пользователя.
cards	ARRAY	["a1a2a3a4", "b1b2b3b4"]	Массив VARCHAR идентификаторов в виде шестнадцатеричных строк (HEX).
fingers	ARRAY	[]	Массив VARCHAR отпечатков пальцев.
faces	ARRAY	[]	Массив VARCHAR лиц.
access	ARRAY	[{}]	Массив JSON с разрешёнными доступами по каналам.
channel	INTEGER	1	Номер разрешенного канала.
schedule	INTEGER	2	Номер расписания доступа для канала.
<pre>{ "count": 1, "persons": [{ "id": 1, "cards": ["a1a2a3a4", "b1b2b3b4"], "fingers": [], "faces": [], "access": [{ "channel": 1, "schedule": 2 }, { "channel": 2, "schedule": 2 }] }] }</pre>			

5.2 Добавление пользователей

Метод позволяет загружать в контроллер одного или нескольких пользователей, включая их уникальные идентификаторы и права доступа. Данные передаются в виде массива объектов, каждый из которых описывает одного пользователя. Идентификаторы карт доступа указываются в поле "cards" как массив шестнадцатеричных строк (HEX). Номер расписания доступа для канала ссылается на "id" годового расписания.

Запрос:

PUT <i>https://<ip_адрес>:8443/api/access/persons</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	persons	Загрузить пользователей.
persons	ARRAY	[{},{}]	Массив JSON пользователей.
id	INTEGER	1	Уникальный номер пользователя.
cards	ARRAY	["a1a2a3a4", "b1b2b3b4"]	Массив VARCHAR идентификаторов в виде HEX.
fingers	ARRAY	[]	Массив VARCHAR отпечатков пальцев.
faces	ARRAY	[]	Массив VARCHAR лиц.
access	ARRAY	[{},{}]	Массив JSON с разрешёнными доступами по каналам.
channel	INTEGER	1	Номер разрешенного канала.
schedule	INTEGER	2	Номер годового расписания для канала.

```
{
  "target": "persons",
  "persons": [
    {
      "id": 1,
      "cards": [
        "c1c2c3c4",
        "d1d2d3d4"
      ],
      "fingers": [],
      "faces": [],
      "access": [
        {
          "channel": 1,
          "schedule": 2
        },
        {
          "channel": 2,
          "schedule": 2
        }
      ]
    }
  ]
},
```

```

    {
      "id": 2,
      "cards": [
        "e1e2e3e4",
        "f1f2f3f4"
      ],
      "fingers": [],
      "faces": [],
      "access": [
        {
          "channel": 1,
          "schedule": 1
        },
        {
          "channel": 2,
          "schedule": 1
        }
      ]
    }
  ]
}

```

Ответ:

Поле	Тип	Значение	Описание
error	INTEGER	0	Все пользователи успешно добавлены.
<pre> { "error": 0 } </pre>			

Ответ с ошибкой:

Поле	Тип	Значение	Описание
error	INTEGER	1	Количество не добавленных пользователей.
persons	ARRAY	[2]	Массив номеров пользователей с ошибкой.
<pre> { "error": 1, "persons": [2] } </pre>			

5.3 Удаление пользователей

Метод позволяет удалять пользователей целиком вместе с их уникальными идентификаторами и правами доступа или удалять только отдельные идентификаторы доступа. Уникальные идентификаторы пользователей для удаления передаются в поле "persons" в виде массива. Идентификаторы карт

доступа указываются в поле “cards” как массив шестнадцатеричных строк (в формате HEX). Удалив все карты у пользователя, он удаляется. В ответе возвращается количество ошибок, возникших при удалении пользователей или карт доступа.

Запрос:

DELETE <i>https://<ip_адрес>:8443/api/access/persons</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	persons	Загрузить пользователей.
persons	ARRAY	[1,2]	Массив INTEGER уникальных номеров пользователей для удаления.
<pre>{ "target": "persons", "persons": [1,2] }</pre>			

Ответ:

Поле	Тип	Значение	Описание
error	INTEGER	0	Количество пользователей с ошибкой на удаление.
<pre>{ "error": 0 }</pre>			

6 Управление и настройка контроллера

Раздел содержит методы для администрирования устройства: управление и настройка точек прохода, сброс данных (пользователей, расписаний, событий) и другие системные операции.

6.1 Управление точками прохода

Метод позволяет удалённо открывать точки прохода и переводить в соответствующие режимы. В ответе возвращается статус выполнения запроса (успех или ошибка).

Запрос на открытие:

POST <i>https://<ip_адрес>:8443/api/access/command</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	open	Открыть дверь.
channels	ARRAY	[1,2]	Массив INTEGER точек проходов.
direction	INTEGER	0	Указание направления прохода 0 или 1 для плат сопряжения с турникетом (необязательное поле).
<pre>{ "target": "open", "channels": [1,2], }</pre>			

```

    "direction": 0
  }

```

Ответ:

Поле	Тип	Значение	Описание
error	INTEGER	0	Наличие ошибки при выполнении. 0 - успешно выполнено. 1 - ошибка при выполнении.
<pre> { "error": 0 } </pre>			

6.2 Установка режима прохода

Запрос на перевод режима прохода: нормальный/блокировка/свободный:

POST <i>https://<ip_адрес>:8443/api/access/command</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	mode	Установить режим прохода для канала.
channels	ARRAY	[1]	Массив INTEGER каналов (точек проходов).
type	INTEGER	2	0 – Нормальный проход (по аутентификации); 1 – Проход заблокирован (запрещен); 2 – Свободный проход (замок открыт).
<pre> { "target": "mode", "channels": [1], "type": 2 } </pre>			

Ответ:

Поле	Тип	Значение	Описание
error	INTEGER	0	Наличие ошибки при выполнении. 0 - успешно выполнено. 1 - ошибка при выполнении.
<pre> { "error": 0 } </pre>			

6.3 Очистка контроллера

Методы позволяют полностью очистить данные, обрабатываемые контроллером: пользователи, расписания, события и связанные объекты удаляются безвозвратно. В ответе возвращается статус выполнения запроса (успех или ошибка).

6.3.1 Очистка пользователей

Удаляет всех пользователей (включая привязанные карты и права доступа) из памяти контроллера. Операция необратима.

Запрос:

POST <i>https://<ip_адрес>:8443/api/system</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	clear	Очистить.
type	VARCHAR	persons	Удалить всех пользователей.
<pre>{ "target": "clear", "type": "persons" }</pre>			

Ответ:

Поле	Тип	Значение	Описание
error	INTEGER	0	Наличие ошибки при выполнении. 0 - успешно выполнено. 1 - ошибка при выполнении.
<pre>{ "error": 0 }</pre>			

6.3.2 Очистка расписаний

Метод удаляет все расписания, загруженные в контроллер. Расписания, привязанные к правам доступа пользователей, становятся недействительными.

Запрос:

POST <i>https://<ip_адрес>:8443/api/system</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	clear	Очистить.
type	VARCHAR	schedules	Удалить все расписания.
<pre>{ "target": "clear", "type": "schedules" }</pre>			

Ответ:

Поле	Тип	Значение	Описание
error	INTEGER	0	Наличие ошибки при выполнении. 0 - успешно выполнено. 1 - ошибка при выполнении.

```
{
  "error": 0
}
```

6.3.3 Очистка событий

Метод удаляет всю историю событий, сохранённую в памяти контроллера. После выполнения журнал будет пуст. Операция необратима.

Запрос:

POST <i>https://<ip_адрес>:8443/api/system</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	clear	Очистить.
type	VARCHAR	events	Удалить все расписания.
<pre>{ "target": "clear", "type": "events" }</pre>			

Ответ:

Поле	Тип	Значение	Описание
error	INTEGER	0	Наличие ошибки при выполнении. 0 - успешно выполнено. 1 - ошибка при выполнении.
<pre>{ "error": 0 }</pre>			

6.3.4 Очистка логов

Удаляет записи диагностического журнала контроллера, включая сообщения о запуске, ошибках, сбоях питания, сетевых событиях и других внутренних операциях. Не затрагивает журнал событий доступа пользователей.

Запрос:

POST <i>https://<ip_адрес>:8443/api/system</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	clear	Очистить.
type	VARCHAR	logs	Удалить все расписания.
<pre>{ "target": "clear", "type": "logs" }</pre>			

Ответ:

Поле	Тип	Значение	Описание
------	-----	----------	----------

error	INTEGER	0	Наличие ошибки при выполнении. 0 - успешно выполнено. 1 - ошибка при выполнении.
<pre>{ "error": 0 }</pre>			

6.3.5 Полная очистка

Полная очистка включает себя очистку пользователей, расписаний, событий и логов.

Запрос:

POST	<i>https://<ip_адрес>:8443/api/system</i>		
Поле	Тип	Значение	Описание
target	VARCHAR	clear	Очистить.
type	VARCHAR	all	Удалить все расписания.
<pre>{ "target": "clear", "type": "all" }</pre>			

Ответ:

Поле	Тип	Значение	Описание
error	INTEGER	0	Наличие ошибки при выполнении. 0 - успешно выполнено. 1 - ошибка при выполнении.
<pre>{ "error": 0 }</pre>			

6.4 Формат идентификатора карты

Настройка определяет длину считываемого идентификатора карты (в байтах). Поддерживаемые значения: 3, 4, 7 и другие, в зависимости от типа карты и считывателя.

6.4.1 Получение формата идентификатора карты

Запрос:

GET	<i>https://<ip_адрес>:8443/api/access/command</i>		
Поле	Тип	Значение	Описание
target	VARCHAR	card	Получить формат идентификатора карты.
<pre>{ "target": "card" }</pre>			

Ответ:

Поле	Тип	Значение	Описание
card	INTEGER	3	Формат идентификатора карты (3 байта).
<pre>{ "card": 3 }</pre>			

6.4.2 Установка формата идентификатора карты

Параметр задаёт количество байт, считываемых из карты в качестве уникального идентификатора. Используется для корректного распознавания карт разных форматов.

Запрос:

PUT <i>https://<ip_адрес>:8443/api/access/command</i>			
Поле	Тип	Значение	Описание
target	VARCHAR	card	Установить формат идентификатора карты.
card	INTEGER	3	Формат идентификатора карты (3 байта).
<pre>{ "target": "card", "card": 3 }</pre>			

Ответ:

Поле	Тип	Значение	Описание
error	INTEGER	0	Наличие ошибки при выполнении. 0 - успешно выполнено. 1 - ошибка при выполнении.
<pre>{ "error": 0 }</pre>			