

ИСО Тополь API интеграции

Предварительная редакция от 27 Октября 2022г.

1. Введение

1.1 Протокол передачи данных без шифрования

Для обмена информацией (без использования шифрования) с Центральным блоком ИСО Тополь используется протокол HTTP/1.1 в соответствии с [RFC 2616 “Hypertext Transfer Protocol — HTTP/1.1”](https://tools.ietf.org/html/rfc2616) <https://tools.ietf.org/html/rfc2616>

Сервер HTTP может быть отключен настройкой `SET_HTTP_SERVER_HTTP` в файле конфигурации сервера.

1.2 Протокол передачи данных с шифрованием*

Для обмена информацией с Центральным блоком ИСО Тополь** с использованием шифрования канала передачи данных используется протокол HTTPS в соответствии с [RFC 2818 “HTTP Over TLS”](https://datatracker.ietf.org/doc/html/rfc2818) <https://datatracker.ietf.org/doc/html/rfc2818>

Сервер HTTPS может быть отключен настройкой `SET_HTTP_SERVER_HTTPS` в файле конфигурации сервера.

* Зашифрованный канал является отдельным модулем и может отсутствовать в ряде комплектов поставки.

** Для корректной работы может потребоваться дополнительная настройка центрального блока ИСО Тополь.

1.3 Формат передачи данных

Данные между Центральным блоком ИСО Тополь и клиентским программным обеспечением передаются в двухстороннем порядке по запросу клиента. Центральный блок не инициирует передачу данных самостоятельно, а только отвечает на запросы клиента. По умолчанию, переменные запроса могут, на усмотрение клиента, передаваться как в самой строке запроса (может быть проще в реализации и удобнее в отладке, но менее безопасно) так и в заголовках запроса (безопасней т.к. в случае использования зашифрованного канала, заголовки шифруются вместе с остальными данными), это поведение может быть изменено настройкой `SET_HTTP_SERVER_HTTPS_ALLOW_GET_ARGS` в файле конфигурации сервера, позволяющей отключить приём переменных в строке запроса.

В случае, когда запрос или ответ требует передачу объекта (или массива объектов), набор передаваемых данных представляет из себя JSON в соответствии с [rfc 8259 «The JavaScript Object Notation \(JSON\) Data Interchange Format»](https://tools.ietf.org/html/rfc8259)

<https://tools.ietf.org/html/rfc8259>

1.3.1 Формат передачи данных о дате и времени

Данные поля типа дата (например дата события) передаются в формате **POSIX time (Unix epoch) + три регистра для миллисекунд** т.е. например дата 1519220475466 = 1519220475 POSIX (21 Февраля 2018 13:41:15 GMT) + 466 миллисекунд = 21.02.2018 13:41:15:466 GMT

1.3.2 Формат передачи идентификаторов

Данные типа UUID представляют из себя уникальный идентификатор, сформированный в соответствии с **RFC 4122 «A Universally Unique Identifier (UUID) URN Namespace»** <https://tools.ietf.org/html/rfc4122>

Передаются UUID в строковом виде в формате, описанном в пункте **3.Namespace Registration Template**, «The formal definition of the UUID string representation»

1.4 Адреса запросов (точки доступа)

Адрес запроса формируется из указателя протокола `http(s): //` + IP адреса центрального блока + стандартного разделителя «:» + номер порта, указанного в конфигурации центрального блока (по умолчанию 8765 для HTTP и 33000 для HTTPS) + тело команды + переменные, перед именем первой переменной используется символ «?» перед последующими «&» Пример запроса к центральному пульту с IP адресом 192.168.0.68:

`http://192.168.0.68:8765/json/getinputsrawvalues?lastEvent=0&inputUuid=3cfa823e-098c-4778-910d-6f3190b3c607`

1.4 Статусы выполнения команд

После получения и обработки команды, центральный блок возвращает один из следующих статусов:

Код	Описание
200 (OK)	Команда успешно получена, выполнена и далее последует результат выполнения
400 (Bad Request)	Команда успешно получена но не выполнялась т.к. не верно указаны (или не указаны вообще) необходимые переменные для выполнения команды. Например запрос <code>http://192.168.0.68:8765/json/topol3/scanline?serialport=2&firstAddress=2&lastAddress=122</code> не может быть выполнен т.к. в нем не указана переменная «IP», указывающая на IP адрес линейного блока
401 (Unauthorized)	Команда не была выполнена т.к. для ее выполнения требуются привилегии выше стандартных, а токен авторизации не был предоставлен либо не действителен
403 (Forbidden)	Команда не была выполнена т.к. предоставленный токен авторизации не имеет необходимых привилегий
	Команда или запрошенный файл не найдены.

404 (Not Found)	Возможные причины: ошибка в синтаксисе команды, отличие в версии протокола или запрос к удаленному файлу журнала событий или данных подсистемы фото/видео верификации событий
500 (Internal Server Error)	Произошла ошибка во время выполнения команды, при этом создается запись в журнале ERRATA
501 (Not Implemented)	Запрошенная команда не поддерживается данной версией программного обеспечения либо отсутствуют необходимые модули для выполнения команды (например без модуля СКУД все команды управления ключами и разделами доступа будут возвращать ошибку 501 до момента подключения модуля)

1.5 Заголовки ответа

К ответу прикрепляются следующие заголовки:

Заголовок	Описание
Content-Type	MIME Тип содержимого в ответе, для JSON части протокола = «application/json; charset=utf-8»
Content-Length	Количество байт в ответе. Указывается всегда при статусе = 200, по этому заголовку можно ориентироваться сколько байт необходимо дождаться от центрального пульта в случае запроса больших массивов по медленным сетям передачи данных.

1.6 Виды запросов и ответов

- запросы начинающиеся на `/json` – в результате выполнения этих запросов центральный блок возвращает набор данных, например запрос событий, устройств, пользователей и т.п. Набор данных представляет из себя JSON в соответствии с [rfc 8259 «The JavaScript Object Notation \(JSON\) Data Interchange Format»](https://tools.ietf.org/html/rfc8259)
<https://tools.ietf.org/html/rfc8259>
- запросы начинающиеся на `/cmd` – в ответ блок возвращает результат выполнения команды, например добавление нового прибора, отметка тревоги обработанной и т.п. Результатом выполнения команды может быть:
 - строка с содержанием **OK** - если команда выполнена успешно и не возвращает дополнительных данных
 - строка с содержанием результата выполнения команды (формат ответа в описании соответствующей команды)
 - строка с содержанием **ERROR** - если команда не была выполнена и в настройках сервера отключена расшифровка ошибок в содержимом ответов (настройка `SET_SEND_HTTP_ERRORS_DEBUG` файла конфигурации)
 - строка с расшифровкой ошибки, если в настройках сервера включена расшифровка

2. Переменные запросов

Имена переменных в запросах унифицированы между всеми запросами и командами, и перечислены в таблице:

Переменная	Написание переменной в запросе
JSON_PARAM_LAST	last
JSON_PARAM_NAME	name
JSON_PARAM_IP	ip
JSON_PARAM_TYPE	type
JSON_PARAM_UUID	uuid
JSON_PARAM_LIMIT	limit
JSON_PARAM_FROM_UUID	fromUuid
JSON_PARAM_FROM_DATE	fromDate
JSON_PARAM_DATE	date
JSON_PARAM_TO_DATE	toDate
JSON_PARAM_VALUE	value
JSON_PARAM_PORT_NAME	portName
JSON_PARAM_ADDR_FROM	addrFrom
JSON_PARAM_ADDR_TO	addrTo
JSON_PARAM_JSON	json
JSON_PARAM_MD5	md5
JSON_PARAM_PATH	path
JSON_PARAM_LOGIN	login
JSON_PARAM_PASSWORD	password
JSON_PARAM_TOKEN	token
JSON_PARAM_NUM	num
JSON_PARAM_MAX	max
JSON_PARAM_MIN	min
JSON_PARAM_CHANNEL	chan

JSON_PARAM_TIME	time
JSON_PARAM_SUBCODE	subcode
JSON_PARAM_DESCRIPTION	description

Для создания списка типовых имён можно использовать следующий фрагмент кода при инициализации переменных:

```
public final static String JSON_PARAM_LAST = "last";
public final static String JSON_PARAM_NAME = "name";
public final static String JSON_PARAM_IP = "ip";
public final static String JSON_PARAM_TYPE = "type";
public final static String JSON_PARAM_UUID = "uuid";
public final static String JSON_PARAM_LIMIT = "limit";
public final static String JSON_PARAM_FROM_UUID = "fromUuid";
public final static String JSON_PARAM_FROM_DATE = "fromDate";
public final static String JSON_PARAM_DATE = "date";
public final static String JSON_PARAM_TO_DATE = "toDate";
public final static String JSON_PARAM_VALUE = "value";
public final static String JSON_PARAM_PORT_NAME = "portName";
public final static String JSON_PARAM_ADDR_FROM = "addrFrom";
public final static String JSON_PARAM_ADDR_TO = "addrTo";
public final static String JSON_PARAM_JSON = "json";
public final static String JSON_PARAM_MD5 = "md5";
public final static String JSON_PARAM_PATH = "path";
public final static String JSON_PARAM_LOGIN = "login";
public final static String JSON_PARAM_PASSWORD = "password";
public final static String JSON_PARAM_TOKEN = "token";
public final static String JSON_PARAM_NUM = "num";
public final static String JSON_PARAM_MAX = "max";
public final static String JSON_PARAM_MIN = "min";
public final static String JSON_PARAM_CHANNEL = "chan";
public final static String JSON_PARAM_TIME = "time";
public final static String JSON_PARAM_SUBCODE = "subcode";
public final static String JSON_PARAM_DESCRIPTION = "description";
```

3. Авторизация

В случае, если в комплекте поставки центрального блока включен модуль авторизации и в настройках сервера включено его использование (настройка [SET_HTTP_SERVER_REQUIRE_AUTH](#)) то к каждому запросу, который требует авторизации необходимо добавлять две переменные:

Переменная	Содержание	Обязательно
JSON_PARAM_LOGIN	Имя пользователя	ДА
JSON_PARAM_PASSWORD	Пароль пользователя	ДА

ВАЖНО! Пароль чувствителен к регистру!

Пример выполнения команды

Запрос: `https://192.168.1.168:33000/cmd/reboot?login=Admin&password=Admin`

Ответ: OK

При неудачном выполнении команды:

- Если имя пользователя и/или пароль не прошли проверку будет возвращен статус **401 (Unauthorized)** и сформированы записи в журнале событий:
 - **Если имя пользователя не было найдено в базе данных:** событие `EVENT_CLIENT_USER_AUTHORIZE_FAIL` со значением предоставленной переменной `JSON_PARAM_LOGIN` в описании события
 - **Если не был предоставлен пароль пользователя:** событие `EVENT_CLIENT_USER_AUTHORIZE_FAIL` со значением предоставленной переменной `JSON_PARAM_LOGIN`+ локализованной строкой **"Не предоставлен пароль!"** в описании события
 - **Если не было предоставлено имя пользователя:** событие `EVENT_CLIENT_USER_AUTHORIZE_FAIL` с локализованной строкой **"Не предоставлен логин!"** в описании события
- Если имя пользователя и/или пароль были предоставлены в не распознанном формате будет сформирован ответ **ERROR**

ВАЖНО! Для упрощения, во всех примерах показаны параметры, передаваемые в строке запроса, однако в реальной интеграции мы настоятельно советуем использовать заголовки запроса, по крайней мере для передачи таких чувствительных данных, как имя пользователя и пароль

При обмене данными переменные `JSON_PARAM_LOGIN` и `JSON_PARAM_PASSWORD` необходимо (если включена настройка `SET_HTTP_SERVER_REQUIRE_AUTH`) указывать при каждом запросе к серверу. В таком случае, помимо разграничений прав доступа, появляется возможность идентификации пользователя в журнале событий - все изменения, такие как изменения настроек системы, создание, удаление и редактирование данных, будут маркироваться персональным идентификатором пользователя, который выполнял соответствующую команду.

Например:

Запрос: `https://192.168.1.168:33000/cmd/enableaction?token=01a0b86e-e3bd-42a2-b7ba-e68f7e0dc32a&uuid=36d169b4-1d26-437d-908b-b62f1b059446`

В данном случае, выполняется команда `enableaction` (включение действия) с указанием идентификатора действия (`uuid=36d169b4-1d26-437d-908b-b62f1b059446`) и токена (`token=01a0b86e-e3bd-42a2-b7ba-e68f7e0dc32a`). Если у пользователя, который получил данный токен, достаточно привилегий на включение действия, то команда будет выполнена и в журнал будет записано событие о включении действия данным пользователем, если

привилегии не достаточно то будет возвращен статус **401 (Unauthorized)** и в журнал будет занесено событие отказа выполнения команды пользователя.

ВАЖНО! В дальнейших примерах и описаниях команд, переменная **JSON_PARAM_TOKEN** будет опущена, её поддерживают все команды и её использование одинаково в любых запросах.

ВАЖНО! Даже если настройка **SET_HTTP_SERVER_REQUIRE_AUTH** отключена, переменную **JSON_PARAM_TOKEN** всёравно можно указывать в запросах - при этом сохранится персонификация действий в журнале событий, но задача по разграничению прав доступа ложится на плечи клиентского программного обеспечения.

Срок жизни токена задаётся в настройках сервера в переменной **SET_SERVER_TOKEN_TTL** (в секундах) и по умолчанию равен 1000 (с.). По окончании срока жизни токена, он перестанет приниматься системой авторизации и в ответ на команду с таким токеном, будет сформирована ошибка и запись **EVENT_CLIENT_USER_AUTHORIZATION_EXPIRED_USED** в журнале событий. Разумно получить новый токен до окончания срока жизни старого, продление выданных токенов невозможно.

ВАЖНО! Настройки сервера, включая текущее ограничение срока жизни создаваемых токенов, можно вычитать с помощью данного API

4. Получение списка приборов

Получение списка приборов является первым шагом в реализации интеграции с ИСО Тополь. С помощью данных списка вы получаете полную иерархию приборов, входов, выходов системы, их настройки, текущие статусы и последние значения.

Получение списка всех приборов в системе осуществляется командой: **/json/getdevicelist [JSON_GET_DEVICES]**

Команда не требует указания параметров

Ответ на команду: Дерево приборов в формате **JSON**.

Пример выполнения команды

Запрос: <https://192.168.1.168:33000/json/getdevicelist>

Ответ:

```
{
  "uuid":"aa8ca397-0977-4cb3-9c49-876fc9d6cb8f",
  "name":"Центральный блок",
  "description":"Центральный блок ИСО Тополь",
  "type":15,
  "serialPortName":"logic-0",
  "online":true,
  "skud":false,
```

```
"devices":[
],
"inputs":[
  {
    "actionsLock":false,
    "enabled":true,
    "uuid":"c22c9133-0756-4b1b-afe6-eb85ef55cac7",
    "holdFault":false,
    "name":"Свободный объем диска",
    "metrics":"Mb",
    "address":0,
    "holdValues":150,
    "holdEnergy":150,
    "holdDescreteTime":1,
    "holdDescreteValue":10.0,
    "weight":100,
    "holdValuesPreEvent":150,
    "holdValuesPostEvent":150,
    "nonFaultMin":50.0,
    "nonFaultMax":1.7976931348623157E308,
    "nonAlarmMin":100.0,
    "nonAlarmMax":1.7976931348623157E308,
    "nonPreAlarmMin":-1.7976931348623157E308,
    "nonPreAlarmMax":1.7976931348623157E308,
    "alarm":false,
    "prealarm":false,
    "prealarmDelay":0,
    "prearmDelay":0,
    "prearm":false,
    "arm":true,
    "autoarm":true,
    "plotterColor":-256,
    "fault":false,
    "holdAlarm":10000,
    "options":"",
    "link":true,
    "lastValue":1142146.281472,
    "lastDate":1625761463230,
    "thresholdReachAlarm":1,
    "thresholdPikesLength":1,
    "analyzePikes":10000,
    "pikes":0,
    "pikesLength":0,
    "energyThresholdInterval":0,
    "analyzeEnergy":10000,
    "type":130
  },
  {
    "actionsLock":false,
    "enabled":true,
```



```
"uuid":"71909cc9-b9e1-4348-b525-42e69165ec70",
"holdFault":false,
"name":"Общий объем диска",
"metrics":"Mb",
"address":1,
"holdValues":150,
"holdEnergy":150,
"holdDescreteTime":1,
"holdDescreteValue":1.0,
"weight":100,
"holdValuesPreEvent":150,
"holdValuesPostEvent":150,
"nonFaultMin":50.0,
"nonFaultMax":1.7976931348623157E308,
"nonAlarmMin":100.0,
"nonAlarmMax":1.7976931348623157E308,
"nonPreAlarmMin":-1.7976931348623157E308,
"nonPreAlarmMax":1.7976931348623157E308,
"alarm":false,
"prealarm":false,
"prealarmDelay":0,
"prearmDelay":0,
"prearm":false,
"arm":true,
"autoarm":true,
"plotterColor":-16776961,
"fault":false,
"holdAlarm":10000,
"options":"",
"link":true,
"lastValue":1472675.078144,
"lastDate":1625761463230,
"thresholdReachAlarm":1,
"thresholdPikesLength":1,
"analyzePikes":10000,
"pikes":0,
"pikesLength":0,
"energyThresholdInterval":0,
"analyzeEnergy":10000,
"type":130
}
],
"outputs":[
],
"fault":false,
"controlFault":true,
"link":true,
"ping":2,
"pingAverage":2,
"pingAverageNum":1000,
```

```
"cyclePing":60019,
"cycleAverage":60019,
"lastUpdated":1625761463230,
"linkErrorsConsThreshold":25,
"linkErrorsCons":0,
"linkErrors":0,
"lastError":0,
"inited":true,
"sort":1000,
"minRefreshTime":60000,
"maxRefreshTime":600000
},
{
  "lastUpdate":0,
  "uuid":"6fe5fbff-a9bc-4c26-8ca1-3712e532a2a4",
  "name":"Б0С-6",
  "description":"Блок обработки сигналов Б0С-6",
  "type":1030,
  "serialAddress":0,
  "serialPortName":"/dev/ttyACM3",
  "online":true,
  "skud":false,
  "devices":[

  ],
  "inputs":[
    {
      "actionsLock":false,
      "enabled":true,
      "uuid":"6606368d-3706-4095-b8e7-f1a4ae9c10a7",
      "holdFault":false,
      "name":"ADC1",
      "address":1,
      "holdValues":12000,
      "holdEnergy":150,
      "holdDescreteTime":1,
      "holdDescreteValue":0.0,
      "weight":100,
      "holdValuesPreEvent":150,
      "holdValuesPostEvent":150,
      "nonFaultMin":-1.7976931348623157E308,
      "nonFaultMax":1.7976931348623157E308,
      "nonAlarmMin":-1.7976931348623157E308,
      "nonAlarmMax":1.7976931348623157E308,
      "nonPreAlarmMin":-1.7976931348623157E308,
      "nonPreAlarmMax":1.7976931348623157E308,
      "alarm":false,
      "prealarm":false,
      "prealarmDelay":0,
      "prearmDelay":0,
      "prearm":false,
```

```
"arm":true,
"autoarm":true,
"plotterColor":-65536,
"fault":false,
"holdAlarm":10000,
"options":"","
"lastValue":32693.0,
"lastDate":1625723030000,
"thresholdReachAlarm":1,
"thresholdPikesLength":1,
"analyzePikes":10000,
"pikes":0,
"pikesLength":0,
"analyzeEnergy":10000,
"type":100
},
{
  "actionsLock":false,
  "enabled":true,
  "uuid":"99a141bf-dae2-4310-8d4b-f9f8652135e6",
  "holdFault":false,
  "name":"ADC1 0-2",
  "address":101,
  "holdValues":12000,
  "holdEnergy":150,
  "holdDiscreteTime":1,
  "holdDiscreteValue":0.0,
  "weight":100,
  "lowCut":0.0,
  "hiCut":2.0,
  "cutAverage":true,
  "envelopeAttack":0.01,
  "envelopeRelease":0.5,
  "resolution":32.0,
  "holdValuesPreEvent":150,
  "holdValuesPostEvent":150,
  "nonFaultMin":-1.7976931348623157E308,
  "nonFaultMax":1.7976931348623157E308,
  "nonAlarmMin":-1.7976931348623157E308,
  "nonAlarmMax":1.7976931348623157E308,
  "nonPreAlarmMin":-1.7976931348623157E308,
  "nonPreAlarmMax":1.7976931348623157E308,
  "alarm":false,
  "prealarm":false,
  "prealarmDelay":0,
  "prearmDelay":0,
  "prearm":false,
  "arm":true,
  "autoarm":true,
  "plotterColor":-8355712,
  "fault":false,
```

```

        "lastAlarm":1625722922828,
        "holdAlarm":10000,
        "options":"","",
        "lastValue":2.3633363246917725,
        "lastDate":1625723029960,
        "threshold":1000.0,
        "thresholdMultiplier":5.0,
        "lastThreshold":1625723028797,
        "thresholdRecalc":1000,
        "thresholdInterval":50000,
        "thresholdMin":1000.0,
        "thresholdMax":5000.0,
        "thresholdReachAlarm":1,
        "thresholdPikesLength":1,
        "analyzePikes":10000,
        "pikes":0,
        "pikesLength":0,
        "multiplier":1.0,
        "energyThreshold":0.0,
        "analyzeEnergy":0,
        "type":100
    }
],
"outputs":[

],
"fault":false,
"controlFault":true,
"ping":0,
"pingAverage":0,
"pingAverageNum":1000,
"cyclePing":35,
"cycleAverage":35,
"lastUpdated":0,
"linkErrorsConsThreshold":25,
"linkErrorsCons":0,
"linkErrors":0,
"lastError":0,
"inited":false,
"sort":1000,
"minRefreshTime":0,
"maxRefreshTime":0
}
]

```

ВАЖНО! В примере показан небольшой отрывок реального ответа. В действительности, ответ на запрос данного типа может занимать значительный объем данных и требует определенного времени на подготовку сервером. Для минимизации нагрузки на сервер в случае слишком частого формирования данного запроса (например, если сервер

одновременно опрашивает несколько клиентов) ответ на данный запрос кешируется сервером и данные обновляются не чаще одного раза в секунду

ВАЖНО! Формально, этого запроса достаточно для реализации простого мониторинга состояния системы, однако, куда более правильно реализовывать мониторинг однократно запросив список приборов и далее запрашивая события системы, сопоставляя идентификаторы в событиях с идентификаторами в локальной копии списка приборов.

ВАЖНО! Описание прибора может содержать не весь набор доступных переменных - если переменная не установлена (значение `null`) или не поддерживается конкретным типом прибора - то она не передаётся.

Переменные в описании прибора:

- `uuid` - Уникальный идентификатор прибора (в формате UUID)
- `name` - Наименование прибора (для удобства настройки системы, приборам можно назначать понятные наименования)
- `description` - Описание прибора (для удобства настройки, приборам можно присваивать расширенные описания с пометками и комментариями)
- `type` - Тип прибора

На данный момент поддерживаются следующие типы приборов:

<code>type</code>	Тип прибора	Наименование в SDK
5	НПФ Полисервис Тополь-1 с RS-485	TYPE_TOPOL1
10	НПФ Полисервис Тополь-3 Линейный блок Ethernet с RS-485	TYPE_TOPOL3L
15	НПФ Полисервис Тополь-3 Центральный блок или сервер	TYPE_TOPOL3SERVER
20	НПФ Полисервис Вибродатчик ВД-3	TYPE_VD3
30	НПФ Полисервис Вибродатчик ВД-5	TYPE_VD5
40	НПФ Полисервис Сейсмодатчик СД-3	TYPE_SD3
50	НПФ Полисервис Вибродатчик ДД-1	TYPE_DD1
60	НПФ Полисервис Концентратор входных шлейфов КХ-6 с протоколом Тополь-3	TYPE_KX6T3
70	НПФ Полисервис Концентратор выходных шлейфов ЕХ-6 с протоколом Тополь-3	TYPE_EX6T3
100	НПФ Полисервис Тополь-ВОЛС	TYPE_TOPOLVOLS
101	НПФ Полисервис Тополь-ВОЛС дочерняя зона	TYPE_TOPOLVOLS_ZONE

105	НПФ Полисервис Тополь-ВОЛС 12	TYPE_TOPOLVOLS12
106	НПФ Полисервис Тополь-ВОЛС 24	TYPE_TOPOLVOLS24
107	НПФ Полисервис Тополь-ВОЛС 48	TYPE_TOPOLVOLS48
200	НПФ Полисервис ДГ-4	TYPE_DG4
116	ЗАО НПФ Болид С2000-2 (прямая интеграция)	TYPE_C20002
510	Фортеза Фосфор	TYPE_FORTEZA_PHOSPHOR
1000	Устройство с поддержкой ONVIF	TYPE_CAM_ONVIF
1010	IP камера	TYPE_CAM_IP
1030	НПФ Полисервис Тополь-АЦП	TYPE_ADC
1100	DSSL Trassir Сервер	TYPE_TRASSIR_SERVER
1110	DSSL Trassir Канал	TYPE_TRASSIR_CHANNEL
1120	DSSL Trassir Монитор	TYPE_TRASSIR_MONITOR
1130	DSSL Trassir Раскладка	TYPE_TRASSIR_VIEW
2000	ЗАО НПФ Болид С2000-ПП	TYPE_BOLID_C2000PP
2010	ЗАО НПФ Болид Сигнал-10	TYPE_BOLID_SIGNAL_10
2020	ЗАО НПФ Болид Сигнал-20	TYPE_BOLID_SIGNAL_20
2030	ЗАО НПФ Болид Сигнал-20М	TYPE_BOLID_SIGNAL_20M
2040	ЗАО НПФ Болид С2000-2	TYPE_BOLID_C2002
2050	ЗАО НПФ Болид С2000-4	TYPE_BOLID_C2004
2060	ЗАО НПФ Болид С2000-СП1	TYPE_BOLID_SP1
2070	ЗАО НПФ Болид С2000-БКИ	TYPE_BOLID_BKI
12000	Oracle VM точка входа	TYPE_ORACLE_VM
12010	Oracle\Oracle Cloud Объектное хранилище	TYPE_ORACLE_OST
12020	Oracle\Oracle Cloud Длительное хранилище	TYPE_ORACLE_LST
12030	Oracle\Oracle Cloud Оперативное хранилище	TYPE_ORACLE_SST

- **serialPortName String** - имя последовательного порта, к которому подключен прибор.

Может быть:

- именем физического порта (в том числе реализованного с помощью переходника с USB или PCI) на центральном блоке или сервере (**COM1**, **/dev/ttyUSB0** и т.п.)
- номером физического порта на конвертере интерфейсов Ethernet->RS-485 (или аналогичном)

- номером виртуального порта для группировки и организации очередей опроса логических или виртуальных приборов (*logic-0, logic-1* и т.п.)
- **online Boolean** - показывает включен ли опрос данного прибора. Если значение **true** то сервер будет пытаться опрашивать прибор, если значение **false** то сервер не будет опрашивать прибор

Важно! поле **online** не показывает текущий статус обмена с прибором, для этого существует поле **link**, которое показывает была ли установлена связь с прибором.

- **skud Boolean** - учитывается ли прибор в подсистеме СКУД. Если значение **true** то приборы, поддерживающие хранение ключей, управляемые реле и прочие функции подсистемы СКУД будут доступны для управления в рамках подсистемы. Если значение **false** то приборы будут игнорироваться подсистемой.

Важно! поле **skud** является устаревшим и может быть удалено из последующих версий API

- **devices Device[] (JSON)** - если в иерархии приборов, у данного прибора есть подчиненные приборы, то они будут описаны в данном списке.
- **inputs Input[] (JSON)** - список входов прибора
 - **actionsLock Boolean** - Вход игнорируется в действиях. Полезно, если требуется на время (например по расписанию) исключить вход из действий системы.
 - **enabled Boolean** - Если установить поле в значение **false** то вход будет исключен из всех систем обработки, перестанет формировать события и учитываться в зонах, в которые он входит
 - **uuid String** - Уникальный идентификатор входа
 - **holdFault Boolean** - Если установить значение в **true** то вход не будет переходить в состояние "исправен" (**fault = 'false'**) по результатам автоматической проверки исправности. Для перехода в состояние "исправен" будет необходима прямая команда смены статуса входа. Полезно в случае если требуется "заморозить" плавающую неисправность критически важного или задублированного датчика.
 - **name String** - Наименование входа
 - **metrics String** - Единица измерения для данных входа
 - **address Integer** - Адрес входа (номер шлейфа)
 - **holdValues Integer** - Количество сэплов, значение которых удерживается в буфере в ОЗУ
 - **holdEnergy Integer** - Количество рассчитанных значений энергии входа, удерживаемых в буфере в ОЗУ
 - **holdDiscreteTime Integer** - Временная дискретность регистрации данных
 - **holdDiscreteValue Double** - Амплитудная дискретность регистрации данных
 - **weight Integer** - Весовой коэффициент входа

Важно! В следующих версиях API добавится возможность указывать весовые коэффициенты непосредственно в настройках зон, что позволит одному входу иметь разные веса в разных зонах

- `holdValuesPreEvent Integer` - Сколько значений до события (тревоги или неисправности) необходимо записывать в регистратор данных
- `holdValuesPostEvent Integer` - Сколько значений после события (тревоги или неисправности) необходимо записывать в регистратор данных
- `nonFaultMin Double` - Минимально допустимое входное значение. Если входное значение будет ниже указанного то будет сформировано событие неисправности
- `nonFaultMax Double` - Максимально допустимое входное значение. Если входное значение будет выше указанного то будет сформировано событие неисправности
- `nonAlarmMin Double` - Минимальное нормальное значение не формирующее тревогу. Если входное значение будет ниже указанного то будет сформировано событие тревоги
- `nonAlarmMax Double` - Максимальное нормальное значение не формирующее тревогу. Если входное значение будет выше указанного то будет сформировано событие тревоги
- `nonPreAlarmMin Double` - Максимальное нормальное значение не формирующее предварительную тревогу. Если входное значение будет выше указанного то будет сформировано событие тревоги
- `nonPreAlarmMax Double` - Максимальное нормальное значение не формирующее предварительную тревогу. Если входное значение будет выше указанного то будет сформировано событие тревоги
- `alarm Boolean` - Вход находится в тревоге
- `prealarm Boolean` - Вход находится в предварительной тревоги
- `prealarmDelay Long` - Задержка перед тревогой, минимальная длительность извещения "тревога" от входа, при котором формируется событие "тревога" в системе
- `prearmDelay Long` - Задержка перед взятием на охрану
- `prearm Boolean` - Вход находится в стадии постановки на охрану
- `arm Boolean` - Вход поставлен на охрану
- `autoarm Boolean` - Автоматическая постановка входа на охрану при включении системы
- `plotterColor Integer` - Цвет входа на графиках в плоттере сигналов
- `fault Boolean` - Истина если у входа зарегистрирована неисправность, не имеющая отношения к линии связи. Возникает в случае если прибор сам доложил о неисправности входа либо текущие значение входа находится за пределами нормы.
- `holdAlarm Long` - Время удержания тревоги (в миллисекундах).

Важно! Если значение `holdAlarm = 0` то тревоги не будут завершаться автоматически, а только по команде API. Если значение `holdAlarm > 0` то тревоги будут автоматически завершаться после истечения указанного времени. При этом, тревоги можно досрочно завершить командой API

- `options String` - Дополнительные опции прибора. Набор опций персонален для каждого типа прибора.
- `link Boolean` - Связь со входом установлена и находится в нормальном состоянии
- `lastValue Double` - Последнее значение входного сигнала, полученное входом

- `lastDate Long` - Последнее время получения входного сигнала
- `threshold Double` - Текущий пороговый уровень
- `thresholdMultiplier Double` - Множитель порога (Число, на которое умножается среднее значение сигнала за период `thresholdInterval`)
- `lastThreshold Long` -Время последнего расчёта порога
- `thresholdRecalc Long` - Интервал (в миллисекундах) пересчёта порога (как часто пересчитывается пороговый уровень)
- `thresholdInterval Long` - Интервал (в миллисекундах) расчёта порога (за какое время рассчитывается пороговый уровень)
- `thresholdMin Double` - Минимально допустимый уровень порога. Ниже указанного в этом поле числа, пороговый уровень опускаться не будет
- `thresholdMax Double` - Максимально допустимый уровень порога. Выше указанного в этом поле числа, пороговый уровень подниматься не будет
- `multiplier Double` - Множитель для входящих значений сигнала. Можно использовать как цифровой усилитель или делитель
- `energy Double` - Текущая энергия входа
- `energyThresholdMultiplier Double` - Множитель порога энергии (Число, на которое умножается среднее значение сигнала за период `energyThresholdInterval`)
- `energyThreshold Double` - Текущий пороговый уровень энергии
- `energyThresholdMin Double` - Минимально допустимый уровень порога энергии. Ниже указанного в этом поле числа, пороговый уровень энергии опускаться не будет
- `energyThresholdMax Double` - Минимально допустимый уровень порога энергии. Ниже указанного в этом поле числа, пороговый уровень энергии опускаться не будет
- `energyReachAlarm Double` - Если энергия достигнет указанного в этом поле значения, будет сформировано событие тревоги
- `thresholdReachAlarm Integer` - Количество превышений сигналом порога, для формирования тревоги по входу
- `thresholdPikesLength Integer` - Продолжительность превышений сигналом порога, для формирования тревоги по входу (в количестве сэплов)
- `analyzePikes Long` - Время (в миллисекундах) за которое происходит анализ количества и продолжительности превышений сигналом порога
- `pikes Integer` - Текущее количество обнаруженных превышений сигналом порога
- `pikesLength Integer` - Текущая обнаруженная продолжительность превышения сигналом порога
- `energyThresholdInterval Long` - Интервал расчёта порога энергии
- `analyzeEnergy Long` - Время (в миллисекундах) за которое анализируется и накапливается энергия для входа
- `type Integer` - Тип входа (в актуальных версиях API не имеет большого значения, оставлено для совместимости с существующими интеграциями)
- `outputs Output[] (JSON)` - список выходов прибора
- `fault Boolean` - если значение поля `true` то прибор отмечен как неисправным, в

противном случае неисправностей не выявлено

- **controlFault Boolean** - если данное поле установлено в значение **false** то у прибора отключен контроль неисправностей - поле **fault** не будет переходить в состояние **true**, события неисправности не будут формироваться.
- **link Boolean** - если поле отсутствует в описании прибора то связь с прибором не была установлена (по умолчанию значение **null**), если поле установлено в значение 'true' то связь с прибором установлена и в данный момент нормальна, если поле установлено в значение **false** то связь с прибором в данный момент отсутствует

Важно! Если для вновь созданного прибора поле **link** не переходит в состояние **true** убедитесь что включен опрос прибора (значение **online** выставлено в **true**)

- **ping Integer** - время доступа до прибора при его последнем опросе.
- **pingAverage Integer** - среднее время доступа до прибора при его опросе
- **pingAverageNum Integer** - количество циклов опроса, за которое усредняется время доступа для расчёта переменной **pingAverage**
- **cyclePing Integer** - время циклического опроса прибора
- **cycleAverage Integer** - среднее циклическое время опроса прибора
- **lastUpdated Long** - время последнего удачного опроса прибора в формате **POSIX time (Unix epoch) + три регистра для миллисекунд**
- **linkErrorsConsThreshold Integer** - количество последовательных ошибок обмена с прибором, до выставления **link=fault** и формирования события потери связи с прибором
- **linkErrorsCons Integer** - текущее количество последовательных ошибок обмена (с момента запуска сервера)
- **linkErrors Integer** - общее количество ошибок обмена (с момента запуска сервера)
- **lastError Integer** - последняя ошибка обмена с прибором
- **inited Boolean** - был ли прибор успешно инициализирован после запуска сервера
- **sort Integer** - порядок сортировки прибора в списке приборов
- **minRefreshTime Long** - минимальное время обновления прибора (в миллисекундах). Прибор не будет опрашиваться чаще, чем указано в данном поле. Полезно для приборов, данные с которых нужны реже, чем позволяет скорость их опроса, либо для приборов, которые не поддерживают слишком частые опросы.
- **maxRefreshTimeLong** - максимальное время обновления прибора (в миллисекундах). Если прибор не был обновлен за данный интервал времени, то формируется событие потери связи и поле **link** переходит в состояние **false**

4. Получение оперативного списка событий

Все изменения состояний и настроек элементов системы протоколируются в журнале событий. Регулярно вычитывая новые события, можно поддерживать локальную копию всех данных в актуальном состоянии.

Получение списка событий системы осуществляется запросом: **/json/getevents**
[JSON_GET_EVENTS]

Команда требует минимум два параметра:

Переменная	Содержание	Обязательно
JSON_PARAM_FROM_UUID	Идентификатор события, начиная с которого необходимо получить новые записи	НЕТ
JSON_PARAM_LIMIT	Ограничитель количества событий для выгрузки	НЕТ
JSON_PARAM_FROM_DATE	Время события, начиная с которого необходимо получить записи	НЕТ
JSON_PARAM_TO_DATE	Время события, по которое необходимо получить записи	НЕТ

При запросе событий без указания параметров, будут предоставлена выгрузка буфера событий сервера в ОЗУ (объем буфера указывается в настройке `SET_EVENTS_CACHE` сервера) с момента запуска.

После получения последних событий, новые события стоит запрашивать с указанием параметра `JSON_PARAM_FROM_UUID`, где значением параметра является идентификатор последнего события.

Для получения событий за определенный временной интервал внутри буфера событий используются параметры `JSON_PARAM_FROM_DATE` и `JSON_PARAM_TO_DATE` которые можно использовать как независимо так и совместно.

Ответ на команду: Список событий в формате **JSON**.

Пример выполнения команды

Запрос: `https://192.168.1.168:33000/json/getevents`

Ответ:

```
[
  {
    "uuid":"e848dc75-e9e4-4564-9725-1354b072612d",
    "date":"1625824181714",
    "code":"101",
    "level":"200",
    "description":"Автоматическое завершение тревоги",
    "input":"a49ed67a-fb9c-4f1a-9f4f-d3046f87fd58"
  },
  {
    "uuid":"aba4aad9-a712-4380-a283-98c86155a27d",
    "date":"1625824341970",
    "code":"1015",
    "level":"10",
    "description":"serialAddress=10",
```

```

    "device": "aa8ca397-0977-4cb3-9c49-876fc9d6cb8f",
    "user": "f3e91da0-5f29-4596-bc49-a73aee4be463"
  },
  {
    "uuid": "50fc0a6e-1d04-49b6-9c99-67eac79d5475",
    "date": "1625824341970",
    "code": "1015",
    "level": "10",
    "description": "serialPortName=COM1",
    "device": "aa8ca397-0977-4cb3-9c49-876fc9d6cb8f",
    "user": "f3e91da0-5f29-4596-bc49-a73aee4be463"
  }
]

```

В примере ответа мы видим три события:

- Событие с кодом **101 = Тревога завершена** для входа с идентификатором **a49ed67a-fb9c-4f1a-9f4f-d3046f87fd58**, произошедшее в **1625824181714 = 09.07.2021, 09:49 (GMT)** с описанием причины завершения тревоги (**автоматическое завершение**)
- Событие с идентификатором **aba4aad9-a712-4380-a283-98c86155a27d**, кодом **1015 = Изменена конфигурация прибора**, ссылкой на прибор (для которого вносились изменения в конфигурацию) **device = aa8ca397-0977-4cb3-9c49-876fc9d6cb8f**, указанием внесенных изменений (**serialAddress=10**), указанием пользователя, который внёс данные изменения (**user = f3e91da0-5f29-4596-bc49-a73aee4be463**) и времени события **1625824341970 = 09.07.2021, 09:52 (GMT)**
- Событие с идентификатором **50fc0a6e-1d04-49b6-9c99-67eac79d5475**, кодом **1015 = Изменена конфигурация прибора**, ссылкой на прибор (для которого вносились изменения в конфигурацию) **device = aa8ca397-0977-4cb3-9c49-876fc9d6cb8f**, указанием внесенных изменений (**serialPortName=COM1**), указанием пользователя, который внёс данные изменения (**user = f3e91da0-5f29-4596-bc49-a73aee4be463**) и времени события **1625824341970 = 09.07.2021, 09:52 (GMT)**

Важно! Обратите внимание что точность определения времени событий - 1 миллисекунда, что не всегда позволяет вести опрос новых событий ориентируясь только на время последнего события в локальной копии (с использованием только параметра **JSON_PARAM_FROM_DATE**) - именно по этому рекомендуется использовать запрос событий с использованием точного указания последнего известного события в параметре **JSON_PARAM_FROM_UUID**

Событие может иметь следующие поля (пустые поля не передаются):

- **date Long** - Дата и время события;
- **module Integer** - Модуль, сформировавший событие*
- **code Integer** - Код события**
- **description String** - Описание события
- **level Integer** - Уровень события***

- **uuid UUID** - Идентификатор события
- **user UUID** - Пользователь, который вызвал данное событие
- **input UUID** - Вход, который принимал участие в событие
- **output UUID** - Выход, который принимал участие в событие
- **device UUID** - Прибор, который принимал участие в событие
- **zone UUID** - Зона, в которой произошло событие
- **key UUID** - Ключ, который принимал участие в событии
- **links String[]** - Ссылки, прикрепленные к событию

Модули:

Код модуля	Значение
3000	Модуль интернационализации
4000	Модуль SQL сервера
5000	Модуль настроек
7000	Модуль HTTP сервера
8000	Модуль HTTP клиента

Коды событий:

Код события	Значение
80	Авторизован пользователь
100	Тревога
101	Тревога завершена
105	Тревога отмечена обработанной
106	Тревога отмечена ложной
107	Прибор инициализирован
110	Поставлен на охрану
111	Снят с охраны
120	Неисправность
121	Неисправность устранена
130	Установлена связь
131	Связь потеряна
140	Постановка на охрану

141	Отмена постановки на охрану
150	Предварительная тревога
151	Предварительная тревога завершена
200	Дискретный выход включен
210	Дискретный выход выключен
220	Состояние регулируемого выхода изменилось
900	Выявлена установка имитатора устройства в линию!
990	Система запущена
991	Система выключается
999	База данных создана при первом запуске или пересоздана по команде администратора
1000	В конфигурацию добавлен новый прибор
1005	Обновлена конфигурация прибора
1006	Настройки записаны в прибор
1007	Ошибка обновления конфигурации прибора
1010	Прибор удален из конфигурации
1020	Программное обеспечение прибора успешно обновлено
1021	Ошибка обновления программного обеспечения прибора
1100	Запущено сканирование линии
1101	Сканирование линии завершено
1102	При сканировании адреса найден новый прибор
1103	При сканировании адреса найден уже существующий прибор
1104	При сканировании адреса найден замененный прибор
1105	При сканировании адреса прибор не найден
1110	Запущена автоматическая адресация
1111	Автоматическая адресация завершена
1112	Автоматическая адресация успешна
1113	Ошибка автоматической адресации - нет ответа
1114	Ошибка автоматической адресации - ошибка CRC
1115	Ошибка автоматической адресации - одинаковые серийные номера в линии

1120	Принудительная замена адреса по серийному номеру
2000	Активация действия
2010	Деактивация действия
3010	Ошибка синтаксиса в аргументах командной строки
3020	Загружен локализованный перевод приложения
3030	ОШИБКА загрузки локализованного перевода приложения
3100	Удалена папка
3101	ОШИБКА удаления папки!
4000	Загружены настройки SQL
4010	ОШИБКА загрузки настроек SQL!
4020	ОШИБКА SQL!
4030	SQL Сервер подключен
4040	SQL ОШИБКА подключения!
4050	SQL Сервер отключен
4060	SQL ОШИБКА отключения!
4070	SQL Таблица создана
4080	SQL Таблица обновлена или изменена
4090	SQL Таблица УДАЛЕНА
4100	SQL ОШИБКА создания таблицы!
4110	SQL создана база данных
4120	Создана резервная копия базы данных
4130	ОШИБКА создания резервной копии базы данных
4140	Список доступных резервных копий баз данных
4200	SQL ОШИБКА при сохранении события!
5000	Установлен тип операционной системы
5010	Определена локализация
5020	Принудительно выставлена локализация
5030	Настройки загружены
5040	ОШИБКА загрузки настроек

5050	Настройка сохранена
5060	ОШИБКА сохранения настройки!
6000	Инициализирована подсистема событий
6010	Неизвестный код события - разные версии клиента и сервера?
6020	Новое исключение (ошибка) программы
6030	Нет связи с сервером SQL, запись отчета об ошибке на диск
6040	Отчет об ошибке записан на диск
6050	ОШИБКА ЗАПИСИ ОТЧЕТА ОБ ОШИБКЕ НА ДИСК!
7000	HTTP сервер запущен
7010	ОШИБКА запуска HTTP сервера!
7020	HTTP клиент подключен
7030	HTTP запрос
7040	HTTP ответ
7050	HTTP ошибка параметра запроса
7060	HTTP ошибка в запросе
7070	HTTP ошибка обработки запроса
8000	HTTP клиент подключен
8010	HTTP клиент ОШИБКА подключения
8020	Ошибка обработки JSON
9010	Информация о диске
9020	Заканчивается свободное место на диске!
9030	Информация о системных ресурсах
9035	Информация о системных температурах
9040	Температура процессора слишком велика!
9100	Установлена системное время
6005	События загружены из базы данных
7080	Создан новый SSL сертификат
7090	ОШИБКА при создании SSL сертификата!
7100	Запущен широковещательный сервер

7110	ОШИБКА запуска широковещательного сервера!
7095	Не найден SSL сертификат!
1015	Изменена конфигурация прибора
1016	Ошибка в конфигурации прибора!
1017	Настройки записаны в базу данных
1018	Ошибка записи настроек в базу данных!
1030	Добавлен новый вход
1040	Добавлен новый выход
1031	Вход удален
1041	Выход удален
1032	Изменена конфигурация входа
1042	Изменена конфигурация выхода
1050	Устройству добавлено подчиненное устройство
1051	Подчиненное устройство удалено
1052	Изменена конфигурация подчиненного устройства
10000	Добавлен локальный последовательный порт
10010	ОШИБКА открытия локального последовательного порта
10020	Эхо последовательного порта включено
10025	Эхо последовательного порта выключено
10015	ОШИБКА записи в локальный последовательный порт
10016	ОШИБКА чтения из локального последовательного порта
10017	ОШИБКА закрытия локального последовательного порта
10018	ОШИБКА очистки локального последовательного порта
10019	ОШИБКА установки параметров локального последовательного порта
4300	Структура SQL обновлена
4310	ОШИБКА обновления схемы SQL
20010	Вход разрешен
20020	Выход разрешен
20030	Вход по кнопке

20040	Выход по кнопке
20050	Вход запрещен
20060	Выход запрещен
20070	Вход
20080	Выход
30010	Найден неизвестный ключ
30020	Создан новый ключ
30030	Ключ сохранен в базе данных
30040	Ключ сохранен в приборе
30050	Ключ удален из базы данных
30060	Ключ удален из прибора
30000	Ключи загружены из базы данных
5070	Не найден файл с конфигурацией сервера!
5080	Не найден файл с конфигурацией клиента!
5075	Создан новый файл с конфигурацией сервера
5085	Создан новый файл с конфигурацией клиента
5090	Сохранены сетевые настройки устройства
40000	В сети доступна новая версия ПО
40010	В сети нет доступных обновлений ПО
40020	Сервер обновлений не доступен (проверьте подключение к интернету!)
40030	Начата загрузка обновления ПО
40040	Хэш файла обновления ОК
40050	ОШИБКА хэша файла обновления
40060	Обновление успешно завершено
10050	ОШИБКА порта интеграции по Modbus
10060	Интеграция по Modbus запущена успешно
1060	Настройки зоны записаны в базу данных
1061	ОШИБКА записи настроек зоны в базу данных!
1065	Конфигурация зоны в системе изменена

2020	Изменена конфигурация действия
1033	Вход не найден в базе данных - прибор или вход удалены?
1043	Выход не найден в базе данных - прибор или выход удалены?
2030	Вход добавлен в действие
2035	Вход удален из действия
2040	Выход добавлен в действие
2045	Выход удален из действия
1106	ОШИБКА CRC при сканировании линии
1107	ОШИБКА ответа при сканировании линии
1080	Конфигурация контроллера зон изменена
11000	Изменена конфигурация формы
1082	В конфигурацию добавлен контроллер зон
1084	Контроллер зон удален из конфигурации
1086	ОШИБКА конфигурации контроллера зон
100001	Клиент подключен к серверу
100005	Клиент не может подключиться к серверу
100010	Соединение с сервером потеряно!
100100	Клиент не может загрузить видео библиотеку
1180	Сканирование LAN начато
1181	Сканирование LAN завершено
1182	Сканирование LAN: Найден прибор
11100	Изменена конфигурация планов
225	Выход инициализирован
230	ОШИБКА выхода
6011	Файл журнала событий удален
6012	Папка дополнительных данных журнала событий удалена
50080	UPnP порт проброшен
50081	UPnP ОШИБКА проброса порта
100200	Авторизован пользователь

100210	ОШИБКА авторизации пользователя
100220	Продлена авторизация пользователя
100230	Завершена авторизация пользователя
100240	Пользователь пытался использовать просроченную авторизацию
100250	При попытке авторизации использован не существующий токен!
100260	ОШИБКА формата токена авторизации
1026	ОШИБКА в конфигурации прибора
2060	ОШИБКА действия

Уровни событий:

Уровень события	Значение
3	Расширенная отладка
5	Отладочное сообщение
10	Информационное сообщение
50	Предупреждение
100	Отказ
200	Тревога
500	Ошибка
999	Катастрофа

События могут иметь любые промежуточные уровни, значения в таблице позволяют ориентироваться по важности уровня события и настраивать систему трансляции и хранения событий необходимым образом.

Рекомендуется не снижать порог трансляции и хранения событий ниже 50

5. Изменение настроек прибора, входов и выходов

Изменение настроек прибора осуществляется командой: **/cmd/setdevice [CMD_SET_DEVICE]**

Команда требует минимум два параметра:

Переменная	Содержание	Обязательно
JSON_PARAM_UUID	Идентификатор прибора	ДА
Наименование изменяемого поля	Новое значение изменяемого поля	ДА

За одну команду можно изменять значение нескольких полей у одного прибора, передавая

имена и новые значения полей по очереди

Ответ на команду: **Строка ОК**

Пример выполнения команды

Запрос: `https://192.168.1.168:33000/cmd/setdevice?uuid=aa8ca397-0977-4cb3-9c49-876fc9d6cb8f&serialPortName=COM1&serialAddress=10`

Ответ: ОК

В данном примере, у прибора с `uuid=aa8ca397-0977-4cb3-9c49-876fc9d6cb8f` изменяются два параметра - полю `serialPortName` присваивается значение `COM1` а полю `serialAddress` присваивается значение `10`

Важно! Значения строковых переменных, содержащих пробелы и/или специальные символы, необходимо преобразовывать в формат `application/x-www-form-urlencoded` в кодировке `UTF-8`.

Начиная с Java 1.4 это можно осуществить функцией `URLEncoder.encode(СтроковоеЗначение, "UTF-8")`

Важно! Командой `CMD_SET_DEVICE` настраиваются только поля прибора, поля входов и выходов настраиваются отдельными командами

Для изменения настроек входов прибора используется команда: `/cmd/setinput`

[CMD_SET_INPUT]

Команда требует минимум два параметра:

Переменная	Содержание	Обязательно
JSON_PARAM_UUID	Идентификатор входа	ДА
Наименование изменяемого поля	Новое значение изменяемого поля	ДА

За одну команду можно изменять значение нескольких полей у одного прибора, передавая имена и новые значения полей по очереди

Ответ на команду: **Строка ОК**

Для изменения настроек выходов прибора используется команда: `/cmd/setoutput`

[CMD_SET_OUTPUT]

Команда требует минимум два параметра:

Переменная	Содержание	Обязательно
JSON_PARAM_UUID	Идентификатор выхода	ДА
Наименование изменяемого поля	Новое значение изменяемого поля	ДА

За одну команду можно изменять значение нескольких полей у одного прибора, передавая

имена и новые значения полей по очереди

Ответ на команду: **Строка OK**

6. Постановка и снятие с охраны

Постановка и снятие с охраны осуществляются командой: **/cmd/setarm [CMD_SET_ARM]**

Команда требует два параметра:

Переменная	Содержание	Обязательно
JSON_PARAM_UUID	Идентификатор входа или зоны	ДА
JSON_PARAM_VALUE	true для постановки false для снятия	ДА

Ответ: **OK** в случае успеха

7. Принудительная тревога и завершение тревоги

С помощью команды: **/cmd/setalarm [CMD_SET_ALARM]** можно как завершить текущую тревогу по входу или зоне, так и принудительно перевести вход или зону в состояние **Тревога**

Команда требует два параметра:

Переменная	Содержание	Обязательно
JSON_PARAM_UUID	Идентификатор входа или зоны	ДА
JSON_PARAM_VALUE	true для постановки false для снятия	ДА

Ответ: **OK** в случае успеха

8. Зоны

Зоны представляют из себя группы входов, которые можно объединять как по физическому принципу (входы, охраняющие одни и те-же участки периметра) так и по логическому - для формирования развитой логики выдачи извещений о тревоге по сравнению сигналов входов.

8.1 Получение списка зон

Список зоны может быть получен с сервера с помощью команды: **/json/getzoneslist [JSON_GET_ZONES]**

Команда не требует указания параметров

Ответ: Список зон в формате **JSON**.

Пример выполнения команды

Запрос: <https://192.168.1.168:33000/json/getzoneslist>

Ответ:

```
[
  {
    "uuid":"24439735-a82e-4a2f-b621-4985237706fc",
    "type":"1",
    "showInController":"true",
    "name":"Дисковые накопители",
    "inputs":[
      "c22c9133-0756-4b1b-afe6-eb85ef55cac7",
      "71909cc9-b9e1-4348-b525-42e69165ec70",
      "ee980398-21ee-4d49-b81a-609753c961e8",
      "437b2739-19f5-4b01-80c5-eb1c8479622d",
      "955cbc78-dfac-42a1-9fbd-4c13d8a6dd44",
      "7abe5b20-22e8-428e-9d65-6a1085757b61",
      "7f1022ef-b4fb-4e59-8108-11e6e8287d75",
      "a49ed67a-fb9c-4f1a-9f4f-d3046f87fd58"
    ],
    "energyComparator":"3.0",
    "minEnergyToCompare":"500",
    "pikesForSUMAlarm":"0",
    "energyForSUMAlarm":"0",
    "pikesMin":"2",
    "pikesLengthMin":"5",
    "nFromZone":"1",
    "weightForAlarm":"0",
    "alarm":"false",
    "arm":"false",
    "fault":"true",
    "link":"true"
  },
  {
    "uuid":"f1e10356-7968-4232-a46f-11f3f216044e",
    "type":"1",
    "showInController":"true",
    "name":"Здоровье сервера",
    "inputs":[
      "1170f76c-8711-465c-bf9d-0b175eaaa568",
      "15091e61-8517-4b90-93f5-4e409e4c5295"
    ],
    "energyComparator":"3.0",
    "minEnergyToCompare":"500",
    "pikesForSUMAlarm":"0",
    "energyForSUMAlarm":"0",
    "pikesMin":"2",
    "pikesLengthMin":"5",
    "nFromZone":"1",
    "weightForAlarm":"0",
    "alarm":"false",
    "arm":"false",
  }
]
```

```

    "fault":"false",
    "link":"true"
  }
]

```

Описание зоны содержит следующие поля:

- **uuid** **UUID** - Идентификатор зоны
- **type** **Integer** - Тип зоны (*устаревшее, оставлено для совместимости со старыми версиями интеграций, в новых версиях возможно комбинировать любые взаимодействия входов для любых типов зон*)
- **showInController** **boolean** - Отображать ли данную зону в контроллерах зон (*устаревшее, оставлено для совместимости со старыми версиями интеграций, в новых версиях контроллеры сами формируют свои списки зон*)
- **name** **String** - Имя зоны
- **inputs** **UUID[]** - Массив идентификаторов входов, входящих в данную зону
- **energyComparator** **Integer** - Кратность энергий при сравнении для тревоги *
- **minEnergyToCompare** **Integer** - Минимальная энергия для сравнения
- **pikesForSUMAlarm** **Integer** - Сумма превышений для тревоги
- **energyForSUMAlarm** **Integer** - Сумма энергии для тревоги
- **pikesMin** **Integer** - Общее количество превышений сигналом порога по всем входам для формирования извещения "Тревога" (поддерживается начиная с версии 39)
- **pikesLengthMin** **Integer** - Общая продолжительность превышений сигналом порога по всем входам для формирования извещения "Тревога" (поддерживается начиная с версии 39)
- **nFromZone** **Integer** - Кол-во сработавших входов для тревоги
- **weightForAlarm** **Integer** - Вес для тревоги
- **alarm** **Integer** - Текущий статус тревоги
- **arm** **Integer** - Текущий статус постановки на охрану
- **fault** **Integer** - Текущий статус исправности зоны
- **link** **Integer** - Текущий статус наличия связи

* Более подробно параметры настройки зоны описаны в руководстве оператора ИСО Тополь

8.2 Редактирование зоны

Поля зоны могут изменены с помощью команды: **/cmd/setzone [CMD_SET_ZONE]**

Команда требует минимум два параметра:

Переменная	Содержание	Обязательно
JSON_PARAM_UUID	Идентификатор зоны	ДА
Наименование изменяемого поля	Новое значение изменяемого поля	ДА

За одну команду можно изменять значение нескольких полей у одной зоны, передавая имена и

новые значения полей по очереди

Ответ на команду: **Строка ОК**

Пример выполнения команды

Запрос: <https://192.168.1.168:33000/cmd/setzone?uuid=24439735-a82e-4a2f-b621-4985237706fc&energyComparator=5&nFromZone=2>

Ответ: ОК

В данном примере полям `energyComparator` и `energyComparator` для зоны `24439735-a82e-4a2f-b621-4985237706fc` присваиваются значения 5 и 2 соответственно.

Пример выполнения команды с массивами в качестве аргумента

Запрос:

<https://127.0.0.1:33000/cmd/setzone?uuid=24439735-a82e-4a2f-b621-4985237706fc&inputs=%22inputs%22%3A%5B%22c22c9133-0756-4b1b-afe6-eb85ef55cac7%22%2C%2271909cc9-b9e1-4348-b525-42e69165ec70%22%2C%22ee980398-21ee-4d49-b81a-609753c961e8%22%2C%22437b2739-19f5-4b01-80c5-eb1c8479622d%22%2C%22955cbc78-dfac-42a1-9fbd-4c13d8a6dd44%22%2C%227abe5b20-22e8-428e-9d65-6a1085757b61%22%2C%227f1022ef-b4fb-4e59-8108-11e6e8287d75%22%2C%22a49ed67a-fb9c-4f1a-9f4f-d3046f87fd58%22%2C%221170f76c-8711-465c-bf9d-0b175eaaa568%22%2C%2215091e61-8517-4b90-93f5-4e409e4c5295%22%5D>

Ответ: ОК

В данном примере полю `inputs` присваивается значение массива, состоящего из следующих идентификаторов входов: `"inputs": ["c22c9133-0756-4b1b-afe6-eb85ef55cac7", "71909cc9-b9e1-4348-b525-42e69165ec70", "ee980398-21ee-4d49-b81a-609753c961e8", "437b2739-19f5-4b01-80c5-eb1c8479622d", "955cbc78-dfac-42a1-9fbd-4c13d8a6dd44", "7abe5b20-22e8-428e-9d65-6a1085757b61", "7f1022ef-b4fb-4e59-8108-11e6e8287d75", "a49ed67a-fb9c-4f1a-9f4f-d3046f87fd58", "1170f76c-8711-465c-bf9d-0b175eaaa568", "15091e61-8517-4b90-93f5-4e409e4c5295"]`

ВАЖНО! Обратите внимание на синтаксис описания массивов по стандарту [Uniform Resource Identifier](https://en.wikipedia.org/wiki/Uniform_Resource_Identifier) → [Percent-encoding](https://en.wikipedia.org/wiki/Uniform_Resource_Identifier), более подробно можно ознакомиться тут: https://en.wikipedia.org/wiki/Uniform_Resource_Identifier

9. Действия

Действия представляют собой реакции системы на различные события. С помощью действий можно настроить включение релейных выходов при тревоге, запись данных фото-видеоверификации по событиям, активацию тревожного монитора и любые другие команды, доступные в вашем наборе модулей ИСО Тополь.

9.1 Получение списка действий

Список действий может быть получен с сервера с помощью команды: **`/json/getactionslist [JSON_GET_ACTIONS]`**

Команда не требует указания параметров

Ответ: Список действий в формате **JSON**.

Пример выполнения команды

Запрос: <https://192.168.1.168:33000/json/getactionslist>

Ответ:

```
[
  {
    "uuid":"ae1a2da6-18da-47a4-88c6-6624599b393f",
    "name":"Сигнал тревоги",
    "type":1,
    "inputs":[
      "24439735-a82e-4a2f-b621-4985237706fc"
    ],
    "outputs":[
      "d3c613db-31ac-4202-bb03-c0b5cf329399"
    ],
    "disableOnFireStateEnd":true,
    "updateActivatedTime":false,
    "activeFrom":0,
    "activateForTime":0,
    "options":"",
    "activated":false,
    "description":""
    "fire":1,
  },
  {
    "uuid":"a94d5dd4-b275-4704-aca2-a2fc33ed8ffe",
    "name":"Включение охлаждения",
    "type":1,
    "inputs":[
      "15091e61-8517-4b90-93f5-4e409e4c5295"
    ],
    "outputs":[
      "adc19dda-0ef4-476d-bd29-062312d9d41b",
      "cd7cae2d-4220-4b4c-a71c-ba8bee692696"
    ],
    "disableOnFireStateEnd":true,
    "updateActivatedTime":false,
    "activeFrom":0,
    "activateForTime":0,
  }
]
```

```

    "options": "",
    "activated": false,
    "description": ""
    "fire": 1,
  }
]

```

Описание действия содержит следующие поля:

- **uuid** **UUID** - Идентификатор действия
- **type** **Integer** - Тип действия

type	Тип действия	Наименование в SDK
1	Включить выход	ACT_OUTPUT_ENABLE
10	Включить выход на заданное время	ACT_OUTPUT_ENABLE_FOR_TIME_PERIOD
20	Выключить выход	ACT_OUTPUT_DISABLE
30	Включить тревожный монитор и отобразить заданный видео канал	ACT_SHOW_VIDEO_IN_ALARM_MONITOR
40	Поставить на охрану	ACT_ARM
50	Снять с охраны	ACT_DEARM
100	Фотоверификация с сервера Trassir	ACT_TRASSIR_CAPTURE_PHOTO
110	Запись видеоверификации	ACT_VIDEO_RECORD
150	Переключить раскладку монитора оператора Trassir на заданную	ACT_TRASSIR_SHOW_TEMPLATE

ВАЖНО! Список поддерживаемых действий модулей-расширений доступен по запросу в их описании.

- **name** **String** - Имя действия
- **inputs** **UUID[]** - Массив идентификаторов входов и/или зон, события которых приводят к активации действия.

ВАЖНО! В массив **inputs** можно одновременно включать как отдельные входы так и зоны целиком. Если действие активируется зоной, то сохраняются все логические настройки зоны при активации действия

- **outputs** **UUID[]** - Массив выходов, которыми управляет данное действие.
- **disableOnFireStateEnd** **Boolean** - Если поле = **true** то действие прекращается при завершении события, которое инициировало его запуск (например при окончании тревоги, событие запускаемое по тревоге, то-же завершится), если поле = **false** то событие останется активным до наступления другого критерия окончания или ручного

завершения.

- **updateActivatedTime Boolean** - Если поле = **true** то время запуска события будет продлеваться пока активен критерий запуска события и при каждом новом наступлении критерия в процессе активности события, если поле = **false** то время запуска фиксируется при активации. Пример - если событие активируется тревогой на 10 секунд, то если поле = **true**, событие завершится только через 10 секунд после окончания тревоги, при условии что за эти 10 секунд не будет сформировано новой тревоги. Если поле = **false** то событие завершится через 10 секунд после старта.
- **activeFrom Long** - Последнее время запуска события
- **activateForTime Long** - Время, на которое необходимо активировать событие (при использовании обратите внимание на поле **updateActivatedTime**)
- **options String** - Дополнительные параметры действия (персональны для разных типов)
- **activated Boolean** - Указывает активировано ли действие на данный момент
- **description String** - Произвольное описание события
- **fire Integer** - Тип запуска события:

fire	Запуск действия	Наименование в SDK
1	При тревоге	FIRE_ALARM
5	От момента наступления тревоги до завершения тревоги	FIRE_FROM_ALARM_TO_DEALARM
10	При завершении тревоги	FIRE_DEALARM
20	При постановке на охрану	FIRE_ARM
30	При снятии с охраны	FIRE_DEARM
40	При неисправности	FIRE_FAULT
50	При восстановлении из неисправности	FIRE_DEFAULT
60	При восстановлении связи	FIRE_LINK_OK
70	При потере связи	FIRE_LINK_FAIL

9.2 Редактирование действий

Поля действий могут изменены с помощью команды: **/cmd/setaction [CMD_SET_ACTION]**

Команда требует минимум два параметра:

Переменная	Содержание	Обязательно
JSON_PARAM_UUID	Идентификатор действия	ДА
Наименование изменяемого поля	Новое значение изменяемого поля	ДА

За одну команду можно изменять значение нескольких полей у одного действия, передавая имена и новые значения полей по очереди

Ответ на команду: **Строка ОК**

Пример выполнения команды

Запрос: <https://192.168.1.168:33000/cmd/setaction?uuid=a94d5dd4-b275-4704-aca2-a2fc33ed8ffe&updateActivatedTime=true&type=10&options=600000>

Ответ: ОК

В данном примере действие [a94d5dd4-b275-4704-aca2-a2fc33ed8ffe](#) ("Включение охлаждения") будет настроено на включение на заданное время (`type=10`) не менее чем на 10 минут (600000 мс.) после завершения события "Тревога" (`updateActivatedTime=true`)

Полный перечень поддерживаемых команд в SDK:

```
public static final String CMD_CREATE_TOKEN = "/cmd/createtoken";
public static final String CMD_DELETE_DEVICE = "/cmd/deletedevice";
public static final String CMD_SET_DEVICE = "/cmd/setdevice";
public static final String CMD_WRITE_SETTINGS_TO_DEVICE =
"/cmd/writesettingstodevice";
public static final String CMD_APPLY_LAN_SETTINGS =
"/cmd/topol3/applylansettings";
public static final String CMD_SET_INPUT = "/cmd/setinput";
public static final String CMD_SET_OUTPUT = "/cmd/setoutput";
public static final String CMD_SET_SHAPE = "/cmd/setshape";
public static final String CMD_EXECUTE_SCRIPT = "/cmd/executescript";
public static final String CMD_DELETE_ZONE = "/cmd/deletezone";
public static final String CMD_SET_ZONE = "/cmd/setzone";
public static final String CMD_SET_PLAN_ELEMENT = "/cmd/setplanelement";
public static final String CMD_DELETE_PLAN_ELEMENT =
"/cmd/deleteplanelement";
public static final String CMD_DELETE_PLAN = "/cmd/deleteplan";
public static final String CMD_SET_PLAN = "/cmd/setplan";
public static final String CMD_SET_OUTPUT_VALUE = "/cmd/setoutputvalue";
public static final String CMD_SET_ZONE_CONTROLLER =
"/cmd/setzonecontroller";
public static final String CMD_DELETE_ZONE_CONTROLLER =
"/cmd/deletezonecontroller";
public static final String CMD_SET_ACTION = "/cmd/setaction";
public static final String CMD_ENABLE_ACTION = "/cmd/enableaction";
public static final String CMD_DISABLE_ACTION = "/cmd/disableaction";
public static final String CMD_DELETE_ACTION = "/cmd/deleteaction";
public static final String CMD_SET_ALARM = "/cmd/setalarm";
public static final String CMD_SET_ARM = "/cmd/setarm";
public static final String CMD_RESTART = "/cmd/restart";
public static final String CMD_RESTART_LAN = "/cmd/lanrestart";
public static final String CMD_SCAN_LAN = "/cmd/scanlan";
public static final String CMD_REBOOT = "/cmd/reboot";
public static final String CMD_SET_TIME = "/cmd/settime";
public static final String CMD_UPDATE_WEB = "/cmd/webupdate";
```

```
public static final String CMD_SCAN_LINE = "/cmd/scanline";
public static final String CMD_CREATE_DEVICE = "/cmd/createdevice";
public static final String CMD_CREATE_ZONE = "/cmd/createzone";
public static final String CMD_CREATE_ACTION = "/cmd/createaction";
public static final String CMD_CREATE_BKP = "/cmd/createbkp";
public static final String CMD_RESTORE_FROM_BKP = "/cmd/restorefrombkp";
public final static String JSON_GET_EVENTS = "/json/getevents";
public final static String JSON_GET_LOG_FOR_DAY = "/json/getlogforday";
public final static String JSON_GET_LOGS_LIST = "/json/getlogslist";
public final static String JSON_GET_SETTINGS = "/json/getsettings";
public static final String JSON_GET_DEVICES = "/json/getdevicelist";
public static final String JSON_GET_DEVICE_STATES =
"/json/getdevicestates";
public static final String JSON_GET_ZONES = "/json/getzoneslist";
public static final String JSON_GET_CLOUDIPS = "/json/getcloudips";
public static final String JSON_SET_ZONES = "/json/setzoneslist";
public static final String JSON_SET_CLOUDIPS = "/json/setcloudips";
public static final String JSON_GET_PLANS = "/json/getplanslist";
public static final String JSON_GET_ZONE_CONTROLLERS =
"/json/getzonecontrollerslist";
public static final String JSON_GET_ACTIONS = "/json/getactionslist";
public static final String JSON_GET_SHAPES = "/json/getshapeslist";
public static final String JSON_GET_OUTPUTS = "/json/getoutputslist";
public static final String JSON_GET_INPUTS_VALUES =
"/json/getinputsvalues";
public static final String JSON_GET_DIR = "/json/getdir";
public final static String JSON_GET_READY = "/json/getready";
public final static String SYS_SET_SETTINGS = "/sys/setsetting";
```